FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

# Automation of a Continuous Processes Unit

**Guilherme Marques Chainho**

WORKING VERSION

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Professor Paulo Portugal

October 28, 2013

© Guilherme Marques Chainho, 2013

# Abstract


Everyday technology suffers an evolutionary and innovative change to a very high pace. Different equipments and applications are emerging and the existing technologies quickly cease to be updated. This change requires a constant updating by all people, but especially by people who work on these technologies.

Faculty of Engineer of University of Porto, is given immense importance to the quality of education, which reflects the good results in the formation of their students. To continue having these results, an important factor is the constant technological updating of the faculty. Educating engineers, includes not only teach theoretical concepts, but also to demonstrate and experience the concepts learned.

In the Automation laboratory of the Electrical Engineering Department, it was considered important to obtain an educational system, for experiments on the theme of continuous processes control, since there were only available equipments to teach discrete processes control. With this objective, it was partially acquired a *Festo* equipment that contemplates the referenced.

This system consists of three independent modules of water tanks, and can control various system characteristics such as temperature, water flow and level. With this dissertation project it is intended to design all the necessary equipment to complete and adapt the acquired system to the needs of the faculty. To complete this aim, it is necessary to work on two main parts: the hardware and the software of the system.

One of the main requirements of this dissertation project, is to allow the system to be controlled by two different users, a local and a remote. In order to guarantee the operation of the system, for the hardware component, it is necessary to isolate and adapt all electrical signals of the system, so that it can be controlled by both users without the risk of damaging the system.

Regarding the software, it is necessary to create a graphical interface that allows the local user to monitor and control the final system. It is also essential that the local control device run an application that controls all operations of the system and correct them in case of danger situations for the system.

This document also aims to create a guide, so that the students know what experiences can make with the system, in order to train and learn the topics related to the control of continuous processes.

# Resumo


Todos os dias a tecnologia sofre uma mudança evolutiva e inovadora a um ritmo muito acelerado. Diferentes equipamentos e aplicações vão surgindo e rapidamente as tecnologias existentes deixam de ser actuais. Esta mudança exige uma actualização constante por parte de todas as pessoas, mas principalmente por parte das pessoas que trabalham sobre estas tecnologias.

Na faculdade de Engenharia da Universidade do Porto, é dada imensa importância à qualidade de ensino o que reflete os bons resultados na formação dos estudantes. Para isso, um importante factor é a constante atualização tecnológica da faculdade. Educar engenheiros, passa não só por leccionar conceitos teóricos, mas também por demonstrar e experimentar os conceitos aprendidos.

No laboratório de Automação do Departamento de Engenharia Electrotécnica, considerou-se importante obter um sistema de ensino para experiências no tema de controlo de processos contínuos, visto que apenas existiam equipamentos para leccionar sistemas de controlo de processos discretos. Com este objectivo, adquiriu-se parcialmente um sistema da *Festo* que contempla os tópicos referidos.

Este sistema consiste em três módulos independentes de tanques de água, e permite controlar diversas características do sistema, como a temperatura, caudal e nível da água. Com esta dissertação pretende-se projectar todo o equipamento necessário para completar e adaptar o sistema adquirido às necessidades da faculdade. Para isso é necessário trabalhar sobre duas partes principais: o *hardware* e o *software* do sistema.

Um dos requisitos principais do trabalho, é permitir que o sistema possa ser controlado por dois utilizadores diferentes, um local e outro remoto. De forma a garantir o funcionamento do sistem, a nível de *hardware* é necessário isolar e adaptar todos os sinais eléctricos do sistema, de forma a este poder ser controlado por ambos os utilizadores sem risco de danificar o sistema.

Em relação ao *software*, é necessário criar uma interface gráfica que permita o utilizador local monitorizar e controlar o sistema final. É também fundamental que o dispositivo de controlo local, corra uma aplicação que controle todas as operações do sistema e as corrija em caso de situações de perigo para o sistem.

Com este documento pretende-se também criar um guia de utilização do sistema, para que os alunos saibam que experiências podem, com o objectivo de treinar e aprender os tópicos relacionados com o controlo de processos contínuos.

# Aknowledgements

To my supervisor Prof. Paulo Portugal for the support, knowledge and guidance.
To my family for the constant support and trust.
To my friends, from BEST, from my home town, from Porto, for all the hours of listening and for the times of joy.
To Filipa, my best girl, for the endless patience and support.

Guilherme Marques Chainho

*"Challenges are what make life interesting and overcoming them is what makes life meaningful.*
*"*

Joshua J. Marine

# Contents

# List of Figures

# List of Tables

# Abbreviations

| | |
|---|---|
| ADS | Automation Device Specification |
| AI | Analogue Input |
| AO | Analogue Output |
| CPU | Central Processing Unit |
| DEMO | Demonstration |
| DI | Digital Input |
| DO | Digital Output |
| DTC | Dead-Time Control |
| FBD | Function Block Diagram |
| FEUP | Faculdade de Engenharia do Porto |
| HMI | Human Machine Interface |
| IC | Integrated Circuit |
| ID | Identification |
| IL | Instructed List |
| in | Integer |
| LD | Ladder Diagram |
| LED | Light-emitting Diode |
| ms | Milisecond |
| ND | Non Defined |
| No. | Number |
| PID | Proportional-Integral-Derivative |
| PLC | Programmable Logic Controller |
| PWM | Pulse Width Modulation |
| REF | Reference |
| RTD | Resistance Temperature Detector |
| SFC | Sequential Function Chart |
| VB | Visual Basic |

# Chapter 1

# Introduction

This chapter explains the context where this dissertation project is inserted, related with the development and improvement of new tools for the automation laboratory with educational purpose. Following, a description of the primary motivations of this project is presented, derived from the lack of observation and experimental material to teach continuous processes to the students. In the end, a brief description of this report structure is made.

## 1.1 Context

Nowadays the technology is evolving in a highest pace than ever. To keep up with this pace, the technological schools and mainly faculties have to improve not only their education methods but very important also, their technology for educating.

As a faculty of high level education, FEUP is always interested and attentive to improve their technology to allow the students to learn by observation and practice. With this vision in mind, every day new technologies are bought and developed by professors and students to maintain the faculty with the highest quality of education possible.

This dissertation project was designed with the very same purpose, develop and complete a technology that was not available in the Automation department of Electrical and Computers Engineering, to help students easily understand the concepts of continuous processes.

## 1.2 Motivation

In the field of Automation technology, it is extremely important the training of open and closed loop control of discrete and continuous processes as it is one of the current main topics in the area. To provide a complete and high qualified training on this topic, it is required to have practical examples to demonstrate the theory taught during the classes. Only this way students will consolidate the knowledge acquired on the course and relate it with real situations.

In the Automation laboratory there is available a system to demonstrate and control discrete processes; nonetheless, there was the necessity to obtain a practical example of a closed loop control for continuous processes. Taking in consideration this requirement, it was acquired three years ago part of a didactic system from Festo Didactic[1] for the automation of continuous processes. Since this equipment was very expensive, it was decided to acquire only the essential parts and leave the final development for students and professors to complete. For these reasons the following parts were absent: some sensors and actuators, control (hardware and software), signals acquisition, conversion and isolation, interlock[2] and the HMI. Until now, this goal was set aside due to the lack of time and resources of the Automation department, which created the opportunity and motivation for this dissertation project. This development project encompasses the main fields related to the Electrical Engineer degree as: electronics and application development. It is so ideal for a dissertation project as it adds a new technology for the faculty and it also concentrates very important knowledges acquired during the Masters degree.

## 1.3 Structure

The remaining of this dissertation is logically divided into five chapters, with the following content:

**Chapter 2** , "Festo Process Control System", describes minutely the system acquired three years ago from Festo that motivated this project, as the alterations made to adapt the system to the laboratory requirements.

**Chapter 3** , "Problem Analysis", exposes the goals of this dissertation project by explaining all the absent parts of the system and the pretended system functionalities for the laboratory. It also briefly describes the solution proposed for the solution of this dissertation.

**Chapter 4** , "Hardware", presents all the hardware parts to be developed for this project, followed by an extensive explanation of the methodology adapted for the development of the required hardware.

**Chapter 5** , "Software", presents the operation of the software attached with the PLC used to control the physical system. It also describes the application developed to allow the user to manipulate the final system.

**Chapter 6** , "Didactic Experiments" describes the possible problems, using the automation system, that can be proposed to the students in order to understand the concept of continuous processes control. It explains and details the steps to implement the proposed tasks.

[1] Automation Company for Education Solutions
[2] System that prevents critical failures that would damage the machine of happening

**Chapter 7** , "Conclusion", resumes the work done on the dissertation and possible improvements for the future, and provides an analysis over the whole project.

For the correct comprehension of this dissertation, all the chapters shall be read sequentially. Those who have only interest on the possible didactic problems of this automation system, may read only the chapter 6. The problems exposition and solution do not require the comprehension of the system implementation.

Some typographical conventions are used to improve the readability of this document. Pattern names always appear in *italic* and some more important notes are highlighted in **bold**.

# Chapter 2

# Festo Process Control System

This chapter describes the *Festo Process Control System* functionalities, followed by a brief explanation of a closed loop control system and how does this equipment represents a closed loop controlled system of continuous processes.

Three sections are followed with the description of all relevant components of each station individually and their operation processes. It is important to refer that the *Festo* system was only partially acquired. For this reason, these three section are divided in two parts: the first one describes the original *Festo* system and the second part lists and describes the replacement components selected to substitute the absent parts of the system.

On the sixth section it is enumerated the alterations made to the original system in order to adapt it to the defined requirements by the responsible professor, both to have more functionalities and to spend less money.

Lastly it is listed all the equipment available for the final pretended system.

## 2.1 Overview

The Festo Didactic Learning System is designed to train professionals on the field of Automation Technology. Generally only components employed in industry are used to ensure realistic training environment. However, with this System is possible to simulate industrial situations in a laboratory environment which is ideal for teaching university students of Automation in continuous processes control.

The following training contents can be taught by means of the Process Control System: [9]

- Measuring of non-electrical, process technology and control technology variables;
- Extension of measuring chains into closed control loops;
- Assembly, interconnection and commissioning of systems;
- Expansion, modification, commissioning, inspection, maintenance and upkeep of process control equipment and error diagnosis in the event of malfunction.

The Process Control System consists on three stations of water tanks equipped with sensors and actuators that can be assembled in different configurations according to the user intention. Each station can be prepared to control and manipulate different continuous variables by an analogue signal. They are listed below:

- Temperature;
- Flow;
- Level.

With the different sensors and actuators, the system permits the user to make a series of operations that can be used to simulate a closed loop of the water inside the tanks and measure and control the desired variables. The main available operations of the system are:

1. Measure different levels of the water in the containers with discrete signals (e.g. maximum and minimum);
2. Circulate the water between different containers or the same;
3. Enable or prevent the water to enter or exit the containers;

With these operations it is desired that the students can interact with the system to control the analogue variables. To make it possible, the diverse discrete sensors can monitor all the relevant aspects from the system and give all the necessary information to the user by an user interface. Then it is desired that the students alter the actuators values using an individual controller for the devices.

It is important to repeat that this is a system that can be assembled in many different ways. In this chapter it is only explained the general aspect of the system, by explaining each component individually and the alterations made from the original system.

A possible sample of the Process Control System fully equipped and installed is shown in Figure 2.1.

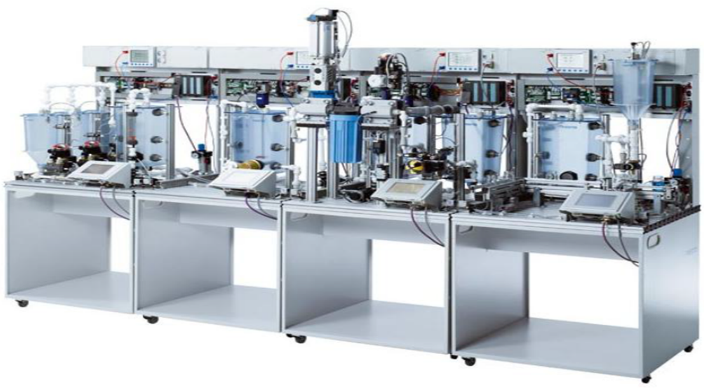

Figure 2.1: A sample of Festo Process Control System fully equipped and installed

Later in this chapter it is explained the original assembly of each station made by Festo and some alterations thought for the laboratory of FEUP.

## 2.2 Stations Methodology

In order to comprehend the operation of the full control system it is simpler to consider the basic configuration of a single-loop control (Figure 2.2).

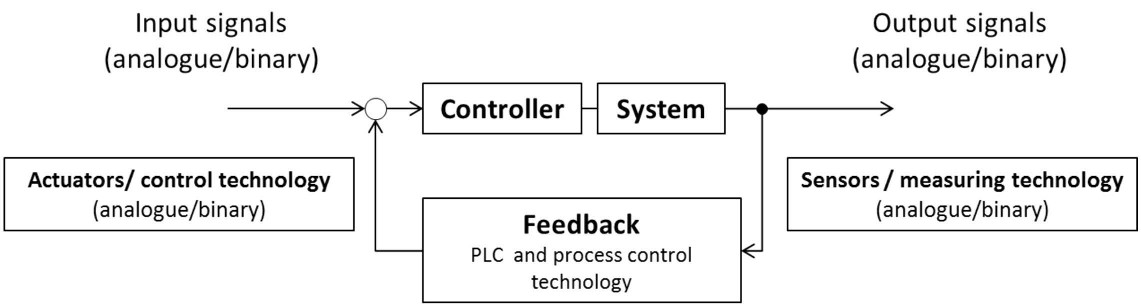


Figure 2.2: Automation system of a closed loop control

As it is shown in the Figure above, each station will receive inputs from the user with the desired values for the actuators. We can consider in each individual station the analogue variables temperature, level or flow, as the interest of this system is to monitor continuous processes and not discrete.

The next step in the process is to make an operation between two signals, the input reference and the output of the feedback. This operation is a subtraction between both signals and the result (error) goes to the controller that will adjust the signal in order to correct it for closer values in relation to the reference.

In the end, the signal will enter corrected in two different places: the final system (in this case the water tanks actuators) and it will be analysed by the sensors to send the feedback to the initial steps. Usually in closed loop systems, and in this specific project, the controller architecture is the PID controller[1]. The loop will be repeated constantly while the system is connected.

This closed-loop is a sample of how the system works in each station, regarding to the variables control. To perfectly understand the full system, it is very important to comprehend not only the operation of the closed-loop control, but also to have a good notion about the physical components that compose each station.

The control of the continuous processes is the main goal of the faculty for this didactic system. It is intended that the students can understand the closed loop control of continuous processes concept by interacting with this system and by controlling the actuators in order to keep the continuous water variables on the pretended values. This package from *Festo* also comes with a manual for the students that have an exposition of the theoretical concepts involving all the system.

[1] **Proportional-integral-derivative** controller is a simple three-term controller that calculates the error between the desired and measured signal and uses an algorithm to estimate and adjust the new input values that will minimize the error

## 2.3 Station 1: Temperature

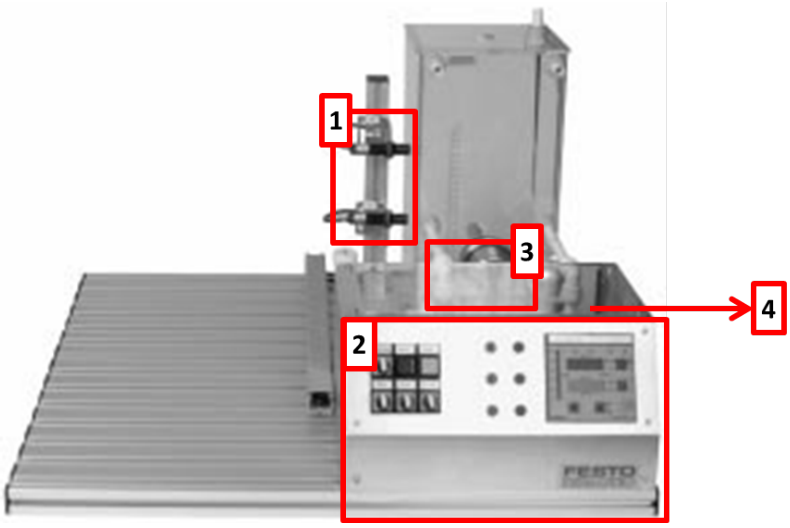


Figure 2.3: Festo temperature control station

The main purpose of this station is to control the temperature of the water in the container using an industrial controller. It follows the same operation as it was described in the previous section and it uses a temperature sensor as a measure device and a heating unit as an actuator.

The physical station is shown in Figure 2.3 and the main components are listed below with their description and relevant specifications.

**1) Capacitive Proximity Sensors**

Figure 2.4: Festo capacitive sensor

**Specifications:**

- Operation Voltage: 10 to 48 V DC
- Switch Output: PNP, Normally open contact
- Maximum switching current: 200 mA
- Current consumption during idling (at 55V): 7 mA

**Function:** These sensors have a capacitor that increases its capacitance when a material with different dielectric from the air approaches. This operation allows the sensors to detect when the water reaches the position closest to the sensor, generating a positive signal to the load output. They can be adjusted in height depending on the user intention.

**2) Controller and control console:** *(Subsequently amended)*

The box numbered with "2" in the Figure 2.3 is composed with two different components. The first one is the control console where all the components for the operation of the station are incorporated. It is by interacting with this console that the user can control every actuator and monitor every sensor (e.g. operating voltage ON or OFF). The second is the controller with a PID algorithm used for the closed loop to control the desired variables (in this station, the temperature).

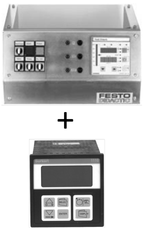

Figure 2.5: Festo Controle console and the controller

**Specifications of the controller:**

- Supply voltage: 24 V AC
- Outputs:
  - Current: 0/4 to 20 mA, maximum load 440Ω
  - Voltage: 0 to 10 V, maximum load current 5 mA
- Inputs:
  - Standard signal (Voltage/Current): 0 to 10 V, 0/4 to 20 mA
  - Frequency:
    * Input signal: 200 mVss to 30 Vss
    * Frequency range: 5 Hz to 900 Hz
    * Signal types: Sinusoidal, square-wave

**3) Temperature Sensor and Heating Unit**

Controlling the temperature is the primary objective of this station and it is necessary to measure and alter this variable as an analogue or digital signal. For this purpose the system has a heating unit (actuator) and a temperature sensor (measurement device).

Figure 2.6: Festo temperature sensor

**Specifications:**

- Resistance designation: PT100
- Measuring range: -50$^{o}$C to approximately +150$^{o}$C
- Output: approximately 60 to 180Ω

**Function:** The platinum resistance temperature detector (RTD) PT100 has normally a resistance of 100Ω at 0$^{o}$C and the resistance value increases as the temperature rises. Figure 2.7 illustrates the variation of the resistance according to the variation of the temperature.

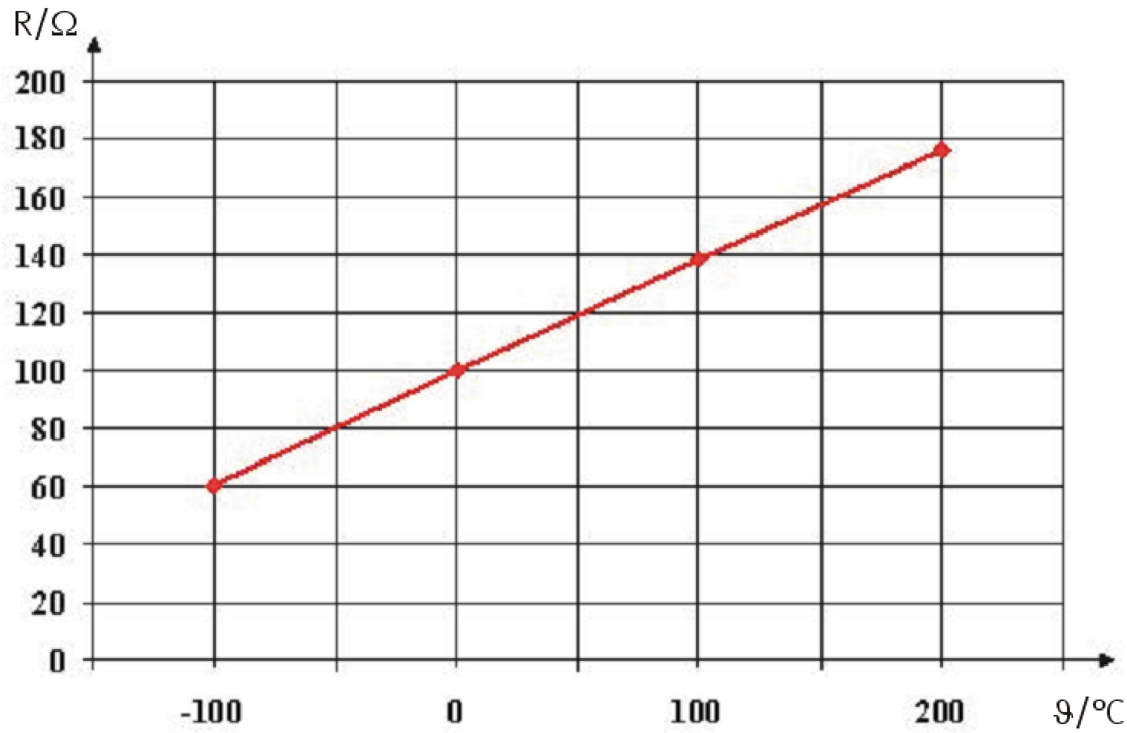


Figure 2.7: Resistance curve of PT100 within a range of -100°C to 200°C [1]

Figure 2.8: Festo heating unit

**Specifications:**

- Supply Power: 230 V AC
- Control Voltage: 24 V DC , 100 mA
- Inputs:
  - Analogue: 0 to 10 V
  - Digital: 24 V DC , 12 mA

**Function:** The heating unit can operate with two different modes - analogue and digital. If the analogue mode is used, the user has to supply 24 V in the digital input and a voltage between 0 and 10 V in the analogue input to control the heating. To select the digital mode, the analogue input has to be supplied by 24 V and the heater is switched OFF or ON by varying the voltage in the digital input with 0 V or 24 V, respectively. These two operation modes are summarized in table 2.1.

| States | Analogue Input | Digital Input |
|---|---|---|
| Digital ON | 24 V | 24 V |
| Digital OFF | 24 V | 0 V |
| Analogue | 0 - 10 V | 24 V |

Table 2.1: Heating unit operation modes

**4) Pump:** *(Subsequently amended)*

The component number four is hidden in figure 2.3. It is a water pump and it allows the station, if desired, to pump the water to the top of the container. It can be used for different means: cool the water, make a disturbance or to create conditions to measure other variables using different sensors (e.g. water flow). Even if it is not critical for the system as it does not influence the acquisition of the temperature variable, it permits to lower or create a disturbance on the temperature, which generates the opportunity to alter the continuous variable and so, change its control process.

Figure 2.9: Festo pump

**Specifications:**

- Operating voltage: 24 V DC
- Power: 26 W
- Current consumption: 0.5 to 0.9 A
- Maximum Flow: 10 l/m
- **Note:** This pump must not operate dry.

### 2.3.1 Components replacements

As explained in the beginning of this chapter, there are some components alterations regarding to the original *Festo* system. The reasons are only related with the high price of the original components, therefore it was decided to search for new, cheaper and equivalent alternatives. The components alterations are listed below:

- **No. 2 -** The Controller and control console **were not acquired** to the system and they are the main reason for this dissertation project. It was decided that due the high price of the components, it would be developed in the faculty a console for all the stations and it was bought one different PID for each continuous variable (figure 2.12);
- **No. 4 -** For the same reason as the console and controller, this specific pump was not acquired and it was substituted by a different one with similar characteristics, shown in figure 2.10.

**4.1) Totton Pump:**

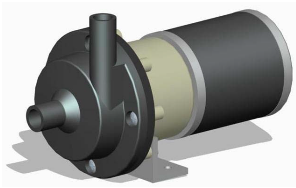

Figure 2.10: *Totton* Pump

**Specifications:**

- Operating voltage: 12 V DC
- Power: 25 W
- Maximum Flow: 15 l/m
- **Note:** This pump must not operate dry.

The *Totton* pump is a normal magnetically coupled centrifugal pump that converts the rotational energy that comes from the electric motor to the hydrodynamic energy of the fluid flow.

The graph of this pump performance is shown in figure 2.11.

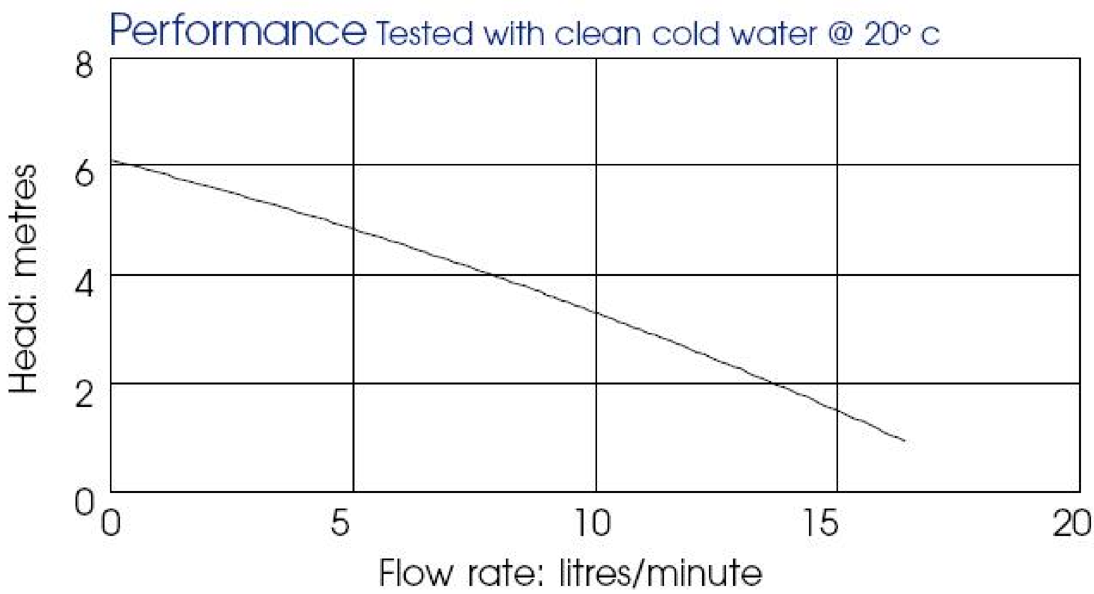


Figure 2.11: Graph performance of the DC Totton pump

**2.1) *Omron* PID:**


Figure 2.12: *Omron* PID

**Specifications:**

- Operating voltage: 100 to 240 V AC, 50/60 Hz (either frequency applicable with same unit)
- Approximately: 12 VA
- Input: Thermocouple (K/J/T/E/R/S/L/U) or platinum resistance thermometer (Pt100/JPt100) selectable
- Ouput: Relay output: open, close; SPST-NO contacts; 3 A, 250 V AC (resistive load); inrush current: 1 A max

The *Omron* PID is specific to control temperatures. It is prepared to receive a signal from the sensor PT100 and to generate an output signal to control the heating unit and alter the temperature of the water to the one chosen by the user.

The description of the PID controls is shown on figure 2.13.

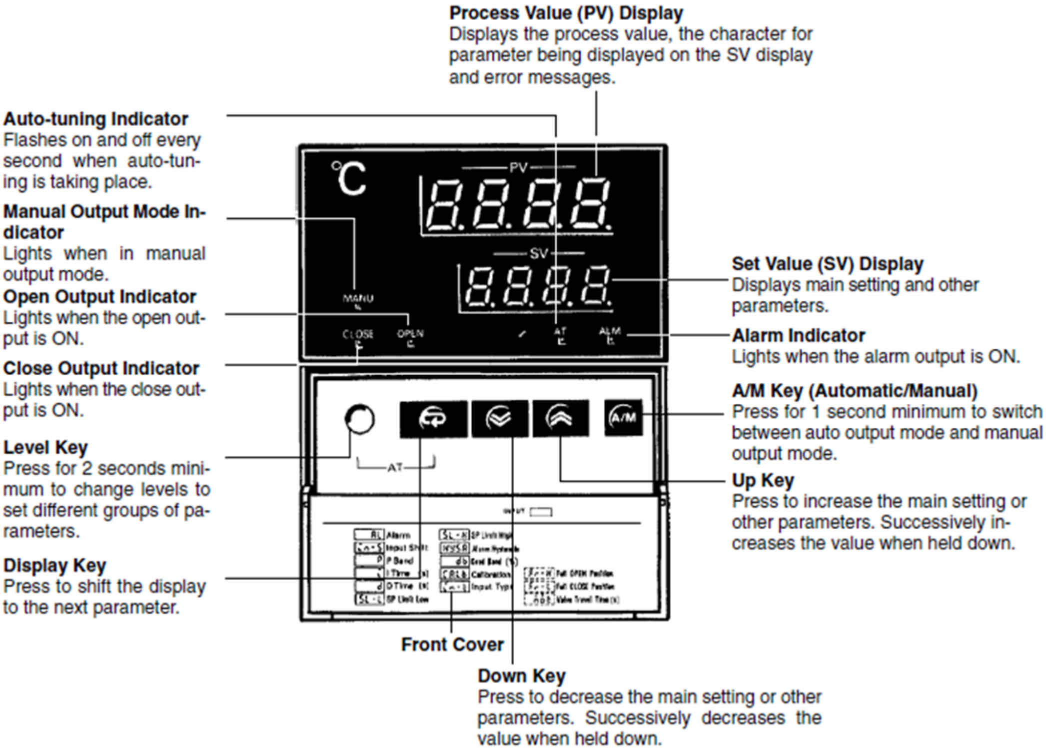


Figure 2.13: PID control operation

## 2.4 Station 2: Flow

The desired variable to control in this station is the flow of the water that goes from the bottom of the container to the top. It is varied by two different modes: by varying the pump speed or/and the proportional valve.

In Figure 2.14 it is only enhanced the components that are different from the previous station.

### 5) Proportional Valve


Figure 2.15: Festo proportional valve

**Specifications:**

- Operation Voltage: 24 V DC
- Power consumption: 8 W
- Input signal: 0 to 10 V

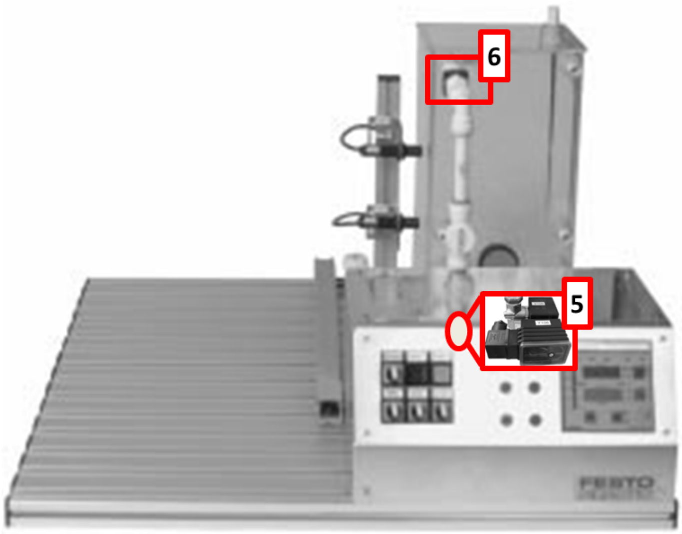


Figure 2.14: *Festo* flow control station

**Function:** To control the flow of the water in the pipes, it is installed a proportional valve that permits to select an opening measure to the valve, given by an analogue signal in voltage or current. Consequently it is possible to create a variation of the flow according to the user aim.

**6) Flow sensor:** *(Subsequently amended)*

Figure 2.16: Festo flow sensor

**Specifications:**

- Operating voltage: 24 V DC
- Current consumption: 18 - 30 mA
- Output (frequency range): 40 - 1200 Hz
- Measuring range: 0.3 - 9.0 l/min

**Function:** When the water crosses the sensor, it creates a rotation on the swirl plate inside the device. Through the optoelectronic infra-red system, it is generated an impulse for each rotation of the swirl. The set of impulses generated will form a square wave with variable frequency, that is the output signal.

There is one more electrical component in this station that is not shown in Figure 2.14 and is essential to create a variation on the water flow. The pump referenced in the previous section has a fixed speed and its purpose is only to pump the water. In this station, to vary the water flow, besides pumping the water in a constant speed, it is required to have the possibility to vary the speed of the pump. For this process the *Festo Didactics* system has available a motor controller

(Figure 2.17).

**7) Motor controller:** *(Subsequently amended)*

Figure 2.17: Festo motor controller

**Specifications:**

- Input:
  - Nominal voltage: 24 V DC
  - Input current: 10 mA
  - Analogue input voltage: 0 to 10 V DC
- Output:
  - Maximum permanent load current: 3.5 A
  - Speed: 0 to $V_{CC}$

Other components used in this station:

- 2 x capacitive proximity sensors
- Control console + controller *(Subsequently amended)*
- Pump *(Subsequently amended)*

### 2.4.1 Components replacements

The alterations at this station are similar to the previous one, with addition to the flow sensor and motor speed controller replacement:

- **No. 4 and No. 2 -** Same conditions as in temperature station;
- **No. 6 -** The flow sensor was replaced for the one shown in figure 2.18;
- **No. 7 -** The motor speed controller, similarly to the control console, is part of the goals of this dissertation project, and it is explained in section 4.5.2.

**6.1) Parker Flowmeter:**


Figure 2.18: Parker flowmeter

**Specifications:**

- Operating voltage: 5 V DC
- Measuring range: 1 - 25 l/min
- Calibration: Typically 752 pulses per Litre
- Output (frequency range - calculated): 12,6 - 314 Hz
- Output voltage (tested): 0,5 V

This flow sensor works exactly in the same way as the *Festo* flow sensor explained before.

## 2.5 Station 3: Level

At this station it is added one more container to allow a water level variation that is measured by an ultrasonic sensor in the second container (Figure 2.19).

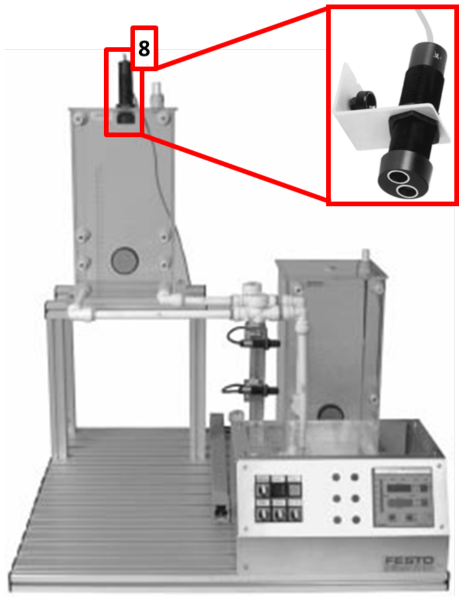


Figure 2.19: Festo level control station

**8) Analogue ultrasonic sensor**

**Specifications:**

- Operation voltage: 24 V DC
- Current output: 4 to 20 mA

**Function:** By generating ultrasonic pulses, the sensor can electronically measure the time that the pulses reflected by the object (water in this situation) take to arrive to the receiver. The output signal of the sensor is proportional to the time that the pulse takes to arrive back to the receiver. Consequently it is possible to obtain an analogue output signal that varies with the distance.

Other components used in this station:

- Pump *(Subsequently amended)*
- Motor Controller *(Subsequently amended)*
- Control console + controller *(Subsequently amended)*
- 2 x Capacitive proximity sensors

### 2.5.1 Components replacement

At this station there is not any new component replacement. There are only the some modifications referred in the other stations: **No. 4** and **No. 5**.

## 2.6 System Adjustments

In order to adapt the system to the professor requirements, it was also acquired some additional components to the system in order to increase its functionalities.

The list of the alterations made in relation to the stations already presented in the previous three sections and respective explanations are:

1. **Installation of three main pumps that fill the containers with water from a main container connected to the three stations -** It can be relevant to alter the quantity of water of each station: create disturbances, renew the water, etc.

2. **Installation of a stirrer and two coolers in the temperature station -** To mix the water inside the container and have all the water at the same temperature. It can also be used to create a disturbance.

3. **Installation of two solenoid valves in each station: one at the entrance of the container and one at the exit -** To improve the control of the system and to create conditions for emergency actions, these two valves on each station are of utmost importance.

4. **Installation of a flow sensor in the level station -** In the level station it is also interesting to analyse the water flow. For this purpose it was acquired one more flow sensor to the system.

5. **Installation of one float switch on each station -** To detect when the container is full, instead of buying one more proximity sensor, it was chosen a float switch also from *Festo* as it is simpler, cheaper and it works for the intended functionality.

6. **Exchange of one proximity sensor in each station for a capacitive level detector -** For the same reason of the previous point.

The specifications and description of the new components are shown below:

**9) Stirrer:**

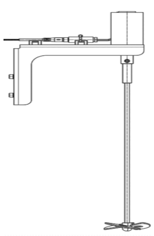

Figure 2.20: Stirrer

**Specifications:**

- Operating voltage: 24 V DC
- Power consumption: 20 W
- Revolution per minute: 3600 - 4200

Figure 2.21: Cooler

**10) Cooler:**

**Specifications:**

- Operating voltage: 12 V DC

**11) Solenoid Valves:**

Figure 2.22: Solenoid Valve

**Specifications:**

- Rated voltage: 12 V DC
- Power consumption: 7 W

This solenoid valve has the particularity that can operate with zero differential pressure, which means that it does not need water inside the pipes or containers to operate.

**12) Float Switch:**

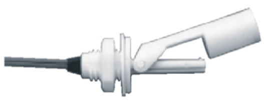

Figure 2.23: Float Sensor

This sensor operates as a switch that passes current or not if the water level drops or rises enough to change the switch position.

**13) Capacitive Level Detector:**

Figure 2.24: Capacitive Level Detector

**Specifications:**

- Operation voltage: 5 to 30 V DC
- Output: DC 100 mA, NPN or PNP
- LED indication when activated

| TEMPERATURE STATION | | | |
|---|---|---|---|
| REF | Figure | Quantity | Components |
| No. 1 | 2.4 | x1 | Capacitive proximity sensor |
| No. 13 | 2.24 | x1 | Capacitive level detector |
| No. 12 | 2.23 | x1 | Float Switch |
| No. 4.1 | 2.10 | x2 | Totton Pump |
| No. 11 | 2.22 | x2 | Solenoid Valve |
| No. 3 | 2.6 and 2.8 | x1 | Temperature sensor + heating unit |
| No. 10 | 2.21 | x2 | Cooler |
| No. 9 | 2.20 | x1 | Stirrer |
| No. 2.1 | 2.12 | x1 | PID |
| **FLOW STATION** | | | |
| No. 1 | 2.4 | x1 | Capacitive proximity sensor |
| No. 13 | 2.24 | x1 | Capacitive level detector |
| No. 12 | 2.23 | x1 | Float Switch |
| No. 4.1 | 2.10 | x2 | *Totton* Pump |
| No. 11 | 2.22 | x2 | Solenoid Valve |
| No. 5 | 2.15 | x1 | Proportional Valve |
| No. 6.1 | 2.18 | x1 | *Parker* Flowmeter |
| No. 2.1 | 2.12 | x1 | PID |
| **LEVEL STATION** | | | |
| No. 1 | 2.4 | x2 | Capacitive proximity sensor |
| No. 13 | 2.24 | x2 | Capacitive level detector |
| No. 12 | 2.23 | x2 | Float Switch |
| No. 4.1 | 2.10 | x2 | *Totton* Pump |
| No. 11 | 2.22 | x4 | Solenoid Valve |
| No. 6.1 | 2.18 | x1 | *Parker* Flowmeter |
| No. 8 | 2.15 | x1 | Analogue Ultrasonic Sensor |
| No. 2.1 | 2.12 | x1 | PID |

Table 2.2: Summary of the final system components

## 2.7 Final System Equipment Summary

To sum up, all the components that are going to be used in the final system are listed on table 2.2.

All the components listed are available already at the laboratory. Only the pumps are partial available, as it was decided to buy only two for testing purposes and the others will be bought on time for the final system assemble.

# Chapter 3

# Problem Analysis

After a good analysis of the system that is available for the dissertation, the most important parts of this project are, first to understand the exact problem to be solved, then, to define a plan of how to attack the problem.

As this system has didactic purposes, it is important to understand that there will be two operation modes of the system: one that is for students usage by a remote mode, other for a local supervision of the system (e.g. professor). Many aspects regarding this dissertation are exactly about the cares to have due to of the possible bad usage of the material by the students.

In the first section of this chapter it is analysed the global problem of this dissertation, followed by the proposed solution.

## 3.1 Problem Exposition

The first part of this dissertation project was to think about the non-existent parts of this system (hardware and software), plus some additional requisites to implement in order to personalise it with the laboratory vision and finally propose a solution. Besides the *Festo* equipment, that permits the reception of signals to control the actuators as the emission of signals that represent the sensors states, it was decided to develop a system that would satisfy the following requirements:

1. **Creation of a physical isolation for the sensors and actuators signals -** As this system is very complexed and expensive and is projected to instruct students on the topic of continuous processes, it is essential to have all the signals isolated, to prevent it from being damaged by the misuse of students that are not used with these technologies.

2. **Implementation of the signals interface between the user and the *Festo* system -** To interact with the sensors and actuators it is required to have a device that reads and writes the users orders to the system and also controls and prioritizes the all the requests.

3. **Signals adaptation -** Not all the signals are with the proper voltage and current for the system specifications, therefore they have to be adapted to the correct requirements.

4. **Development of a HMI to the system -** To compensate the absent of the *Festo* console and to create a better visual interaction with the physical system, it has to be developed a HMI to permit the user to control and monitor the system.

5. **Creation of an interlock -** To prevent the system of being damaged, more than a physical isolation of the signals, it is also necessary to have some protection for wrong orders by the user. In case any condition of the system comes to an extreme (e.g. high temperature of the water, risk of water overflow on the containers, etc), there has to be a monitoring application behind the normal system operation, that stops the actuators and corrects the operation.

With the automation system finalized, it is important to elucidate the students on how to use this system to learn and train continuous processes control. It is so very important to make an explanation and suggestion of what kind of tasks can they do with this system in order to understand the related topics.

## 3.2 Proposed Solution

To complete and fulfil the requirements referred in the previous section, the scheme in Figure 3.1 demonstrates the suggested solution.

As it is shown in the figure 3.1, the solution is divided in five parts. From these parts, only three will be developed in the dissertation by the following order:

1. **PLC[1] (A3) -** This industrial machine has three important functions in this project. First, it can read the different inputs, after being converted from the system, and interacts with the user transforming this signals to digital variables. It also sends the outputs ordered by the user to the hardware section, so that they can be converted and actuated in the final system. The last function is to run the interlock that is constantly monitoring the sensors and actuators and evaluating dangerous situations that can prejudice the system, taking correction measures when necessary.

2. **Hardware (A4) -** Isolate, control and adapt all signals to the system requisites. Not all the sensors or actuators have the same output voltage and current as the electrical interface (PLC) requires. There are also some analogue actuators that need to be controlled and, as it was said before, the controllers from *Festo* are not available. Therefore, it is necessary to implement the hardware to complement the system and allow this operations.

3. **HMI (A1) -** To allow the user to interact locally with the system, it is necessary a HMI to make an intuitive communication with the machine. In this HMI it is possible to monitor or order any change of the variables (inputs and outputs), to verify the alarms occurring in real time and to choose which user is in control of the system, the remote or local.

[1] A Programmable Logic Controller is a digital computer for automation of industrial processes in a real time system. It is designed for multiple inputs and outputs arrangements, extended temperature range, immunity to electrical noise, and resistance to vibration and impact. [10]

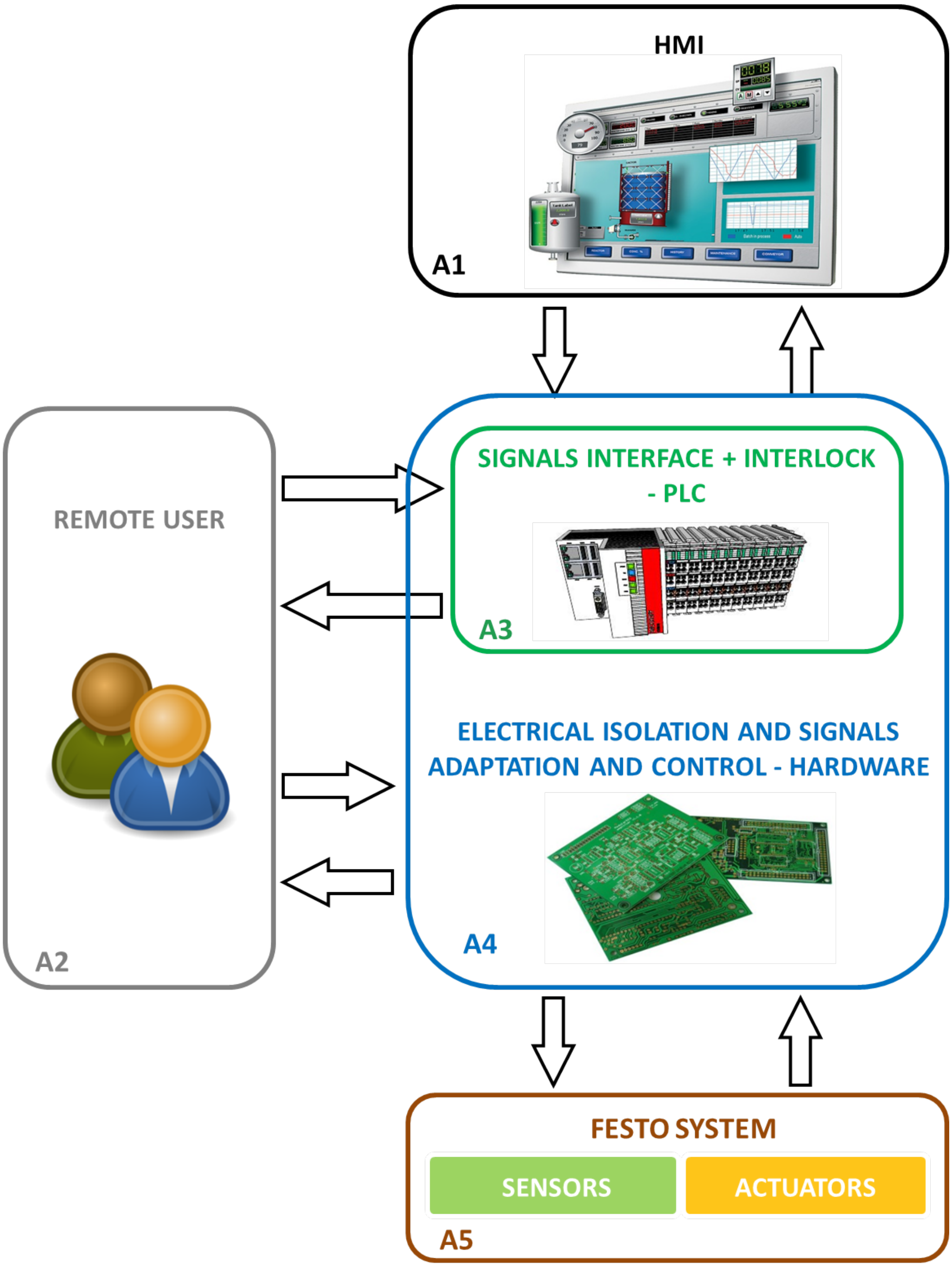


Figure 3.1: General overview of the system to be implemented on this dissertation project

It is important to observe that there are two different possibilities to operate with the system: remotely (A2) and locally (A1). In this dissertation project the local signals are going to be generated through the PLC device (A3), differently from the remote option. The intention is to prepare all signals to be used remotely, but leave this option opened for students experiments. Therefore, the remote signals generation is not part of this dissertation goals. It means that the remote signals are not generated by the PLC, but by a remote device that is not considered in this dissertation.

In figure 3.1 the division A2 is however also connected to the PLC. Why? As there are enough digital input channels available on the PLC, it was opted to connect the digital inputs and outputs to the PLC and the analogue inputs and outputs directly to the remote device. This solution option has 2 advantages:

- Permits to select and control which operation is desired in the PLC via software (locally or remotely);
- Enables to create an interlock by software with no need for many extra hardware. With almost all the variables available in the PLC, it is only necessary to add some extra variables to be able to control even the analogue signals, which permits to stop all signals (remote and local) if necessary.

With this solution there will be only few remote analogue variables that cannot pass through the PLC due to the lack of analogue input and output channels available. This variables are directly connected to the remote device (A2), having a proper control by hardware explained later in chapter 4.

In A5 it is demonstrated the physical components of Festo system. As it was partial acquired and later completed with the replacement components as described in chapter 2, its hardware selection is not included in this dissertation goals or solution.

## 3.3 Conclusion

This chapter pretends to describe the idea to implement in the water tanks system to complete and finalize it. After an intensive analysis of the requirements and the available equipment, it was decided to follow the structure described in the previous section as it fulfils all the requirements mentioned in the beginning of the chapter. Although being slightly more complex and complicated than the original *Festo* system described, besides being extremely cheaper than the first mentioned, it is also more flexible to the teaching necessities thought by the responsible professor. With this solution, the students will be able to test their own experiments while the professor can supervise and control the system locally. Even without the professor supervision, with the interlock running it is totally safe to interact with the system as it is protected both by software and by hardware.

# Chapter 4

# Hardware

The first stated problem in the previous section is the hardware to implement on the automation project. In this chapter, the first two sections expose the general solution for each station hardware and their signals enumeration and specifications, for a better understanding of the system.

Following, in the next two sections, an explanation of the three signals operations needed: isolation, adaptation and control. It is explained the possible solutions and then shown the circuits design as the respective necessary calculations.

## 4.1 Overview

After the analysis of the general solution for this dissertation problem, this chapter is exclusively focused on the hardware part of the solution implementation. The figure 4.1 represents the signals interactions between the remote user, the PLC and finally the automation system. As said in the previous chapter, the requirements for this part of the dissertation are to isolate, adapt and control the required signals. The general case of one station is shown on figure 4.1 and it is going to be explained next in this chapter. Each red square of the figure represent a hardware part to be implemented. However there is still a part of the signals control and adaptation that is not represented on the figure as it is specific of few components. Its explanation is apart of the general description shown in Figure 4.1. These cases are explained later on this chapter and are:

1. **Motor speed control -** The pump of the level station and the stirrer of the temperature station need to have a speed variation controller that is not available due to the absent of the Festo controller component. For this reason it is necessary to develop a new speed variation controller for both motors;

2. **Flow signal analogue conversion -** The flow sensor, as it is described in chapter 2, emits an impulse for each rotation of the swirl plate caused by the water flow. This sensor generates a signal output with fixed voltage and variable frequency. Consequently, it is necessary to convert this frequency variation to a voltage variation, as the remote device will be only prepared to read the signal by a voltage variation. For the local user the flow signal is directly connected to digital input of the PLC and is converted to a digital variable by software;

3. **Flow signal digital conversion -** The PLC digital input channels only accept signals of 24 V and the flow sensor emits an output signal of 5 V. It is so necessary to adapt this signal to the PLC channels requisites.

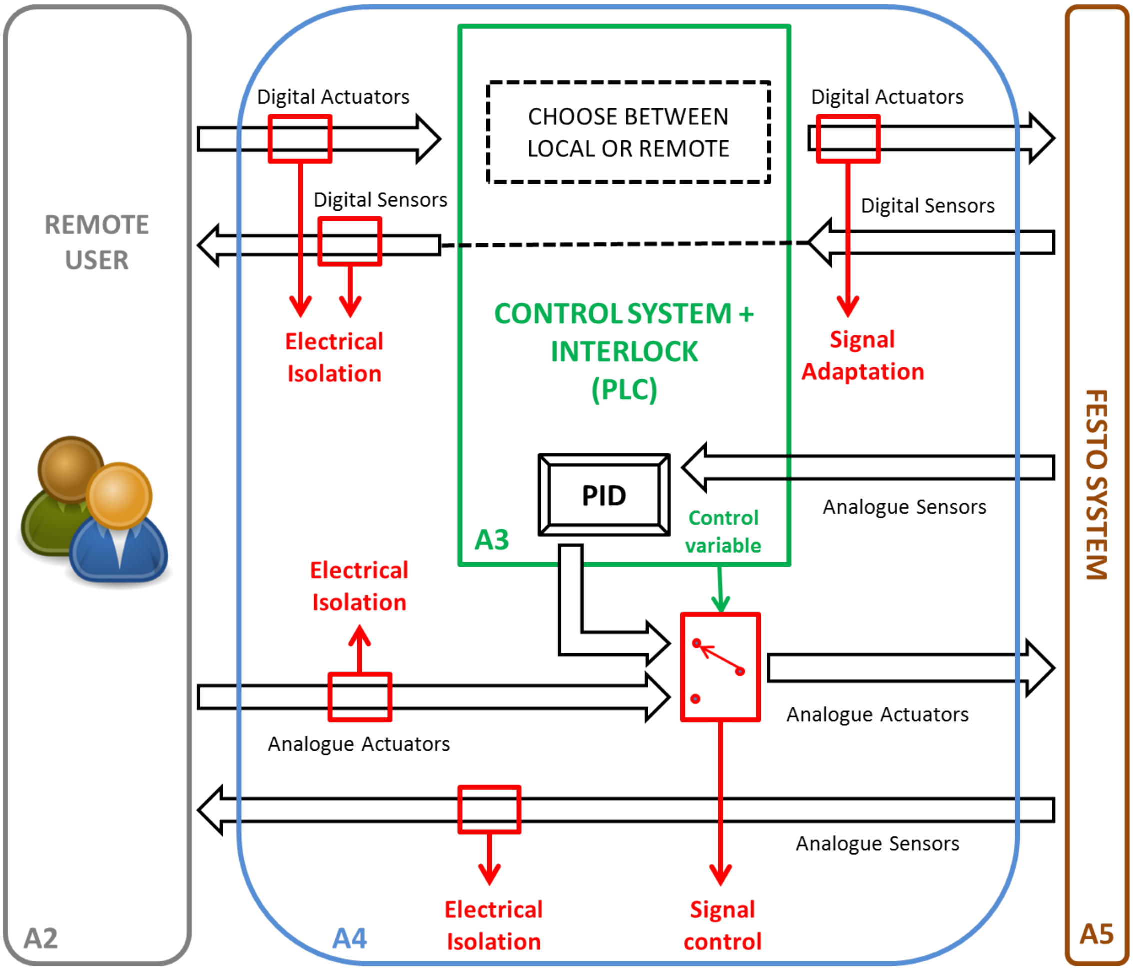


Figure 4.1: Scheme of the hardware solution

From figure 4.1 is still important to make a brief explanation.

As it is possible to observe, the PLC is used for the control of the all system. For that reason, even if the signals do not pass through it, there have to be control signals generated by the PLC to permit the full control of the system.

The hardware development of the system can be divided in two parts: the digital signals and the analogue signals.

**Digital signals:**

In the previous chapter it is explained that all the digital signals generated by the remote user pass through the PLC. This process permits the PLC to have access to the remote and local signals (given by the HMI) and then choose which ones will act with the system (this control part will be done by software and is explained in chapter 5).

As this system is for didactic purposes and also because it is extremely expensive, it is of utmost importance to isolate the signals to avoid the system damage. The two red rectangles on the top left corner of the figure, represent the signals isolation from both parts, from the PLC to the remote device and the other way around.

After having all the digital signals isolated, the PLC generates the chosen signals (between the remote or the HMI values) to the system actuators. However, these signals are not at the correct voltages and currents required by the final actuators. The PLC generates 24 V digital signals and the digital actuators vary between 5 to 24 V. For that reason is necessary to make their adaptation, that is represented by the red rectangle on the top right corner of the figure.

**Analogue signals:**

As the PLC only has four analogue input channels and two output channels, it would not be possible to use the same process as it is used for the digital signals. It was then decided to connect directly the analogue signals to the final actuators, considering the analogue signals from the local user, the PIDs.

The first thing to implement, similarly to the digital signals, was the electrical isolation of the signals (represented by the red rectangles on the bottom left corner).

Finally it was necessary to permit the PLC to control which analogue signals to choose for the final actuators (the remote or the PID). In order to make it, the PLC generates some control digital signals that choose which analogue signals to use by a small hardware control system, represented by the red rectangle on the bottom left side.

It is important to notice that the signals that go directly from the water tank system to the PLC don't need to be isolated as they are already in the correct voltage and current and so, they are not going to be manipulated.

## 4.2 PLC Selection

To monitor and control all the signals necessary for the operation of the system it was opted to use a PLC from Beckhoff. The reason is that there is an available PLC on the laboratory that satisfies all the requirements of the system making this option mandatory by the responsible professor. The table 4.1 represents the important channels for this project, including the available digital and analogue channels. As it will be demonstrated later on this chapter, the channels available suit for the imperative requisites of the system. There are some extra implementations that could be interesting to implement and that will not be because of the lack of channels, however it can always be added later if desired. Some of the ideas left behind due to this limitation are:

- In the local operation, not only control the analogue variables by the PID, but also directly control them by the HMI (increase or decrease their value);

- Monitor all the analogue variables in the HMI (not only the sensors but also the actuators values given by the PID or remotely).

| | Ref. | Items available | No. I/O | Total |
|---|---|---|---|---|
| Digital Inputs (0 or 24 V) | 1418 | 4 | 8 | 32 |
| | 1114 | 2 | 4 | 8 |
| | | | | **40** |
| Digital Outputs (0 or 24 V) | 2114 | 2 | 4 | 8 |
| | 2424 | 4 | 4 | 16 |
| | | | | **24** |
| Analogue Voltage Inputs (0 to 10 V) | 3064 | 1 | 4 | 4 |
| | | | | **4** |
| Analogue Voltage Outputs (-10 to 10 V) | 4032 | 1 | 2 | 2 |
| | | | | **2** |
| Power supply units and I/O interfaces | CX1100-001 | | | **1** |

Table 4.1: PLC inputs and outputs voltage channels available

## 4.3 Signals Analysis

In chapter 2 it is explained the full Festo system operations and modifications and also its sensors and actuators. Now it is important to list and analyse all the signals of each station to comprehend if they need to be isolated or adapted and what type of isolation is required.

**Temperature station:**

In Figure 4.2 is shown the scheme of the sensors and actuators from the temperature automation station with all the alterations regarding to the original assemble (exchange and addition of components).

Some of the components have two numbers on their figure; in this case, the heating unit, the cooler and the stirrer. The heating unit and the cooler are explained in the previous chapter, but to review the respective reasons: one is because it has two input controllers, for analogue or digital control and the other because they are two units, so one actuator for each unit. In the stirrer case the reason is similar to the heating unit. It is of the professor interest to permit the students to control the stirrer by a digital or an analogue signal, so, as it is shown on the signals list, each number corresponds to the digital or analogue control.

On table 4.2 is listed the actuators and sensors from the figure above.

From the signals shown on the table 4.2, there are two that are not available to be controlled by the remote user: *Cooler 1* and *Cooler 2*. These two actuators have the only purpose of creating a faster disturbance by mixing the new water that comes through the pipes, with the water already located inside the container. As this application is for didactic purposes, the only interest to create disturbances is to influence the difficulty of the control of the continuous processes, so only the professors (local user) needs to have this option.

As the PID is the device that generates the local analogue heating unit signal, there is no need for an analogue output channel on the PLC. Therefore the final count of digital and analogue signals channels necessary for this station on the PLC, counting with the remote signals for the actuators that need to pass trough the PLC, are:

- **Analogue signals outputs (AO) -** 1
- **Analogue signals inputs (AI) -** 1
- **Digital signals outputs (DO) -** 9
- **Digital signals inputs (DI) -** 10

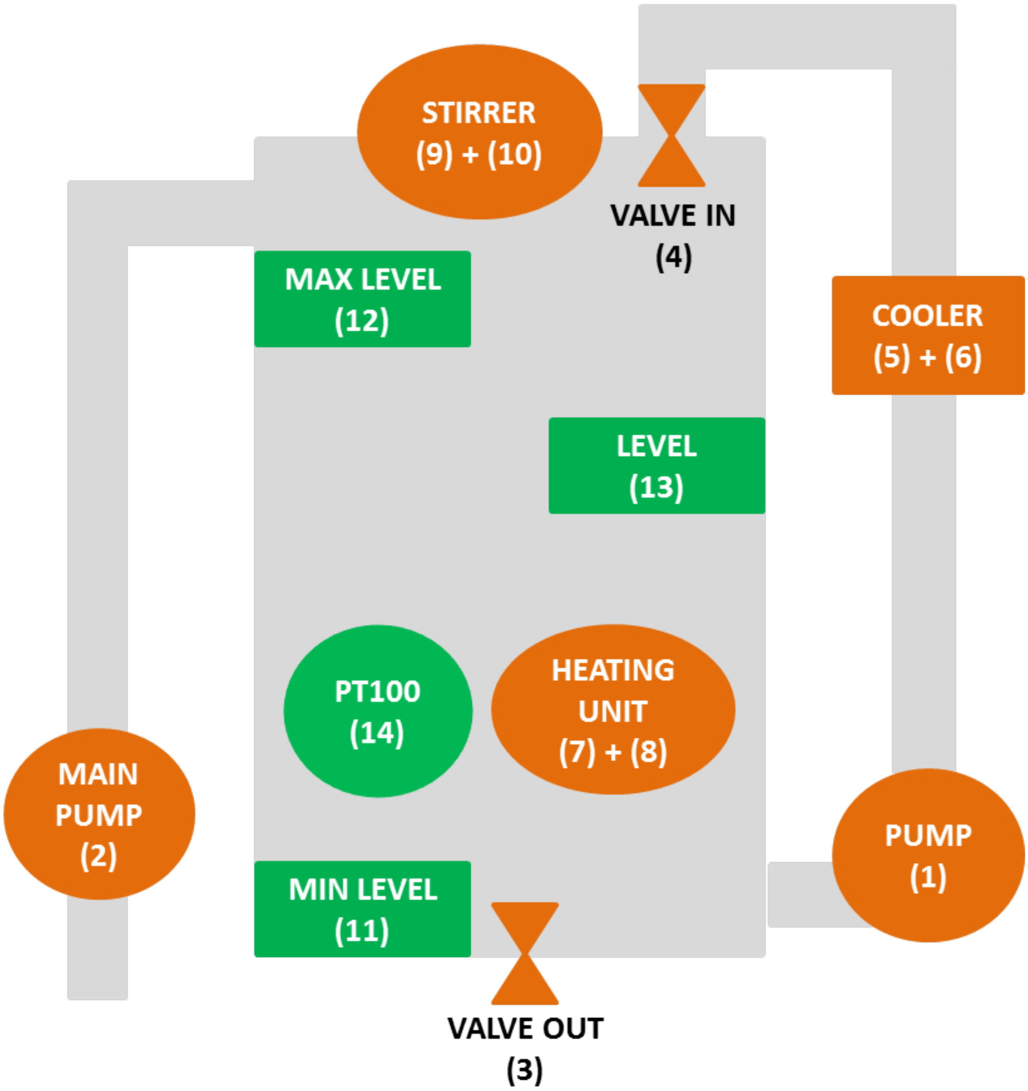


Figure 4.2: Sensors (green) and actuators (orange) available in the Temperature Station

| Ref. | Variable name | Variable type | Operation Voltage | No. Channels |
|---|---|---|---|---|
| | | ACTUATORS | | |
| 1 | Pump | Digital | 0 / 12 V | 1 DI + 1 DO |
| 2 | Main pump | Digital | 0 / 12 V | 1 DI + 1 DO |
| 3 | Output valve | Digital | 0 / 12 V | 1 DI + 1 DO |
| 4 | Input valve | Digital | 0 / 12 V | 1 DI + 1 DO |
| 5 | Cooler 1 | Digital | 0 / 12 V | 1 DO |
| 6 | Cooler 2 | Digital | 0 / 12 V | 1 DO |
| 7 | Heating unit | Digital | 0 / 24 V | 1 DI + 1 DO |
| 8 | Heating unit D/A* | Analogue or Digital | 0 - 10 V or 0 / 24 V | 1 DI + 1 DO |
| 9 | Stirrer Analogue | Analogue | 0 - 5 V | 1 AO |
| 10 | Stirrer Digital | Digital | 0 / 5 V | 1 DI + 1 DO |
| | | SENSORS | | |
| 11 | Minimum Level | Digital | 0 / 24 V | 1 DI |
| 12 | Maximum Level | Digital | 0 / 24 V | 1 DI |
| 13 | Level | Digital | 0 / 24 V | 1 DI |
| 14 | Temperature Sensor - PT100 | Analogue | 0 - 10 V | 1 AI |

Table 4.2: Temperature station sensors and actuators variables

**Level station:**

In Figure 4.3 is shown the scheme of the sensors and actuators from the level automation station with all the alterations regarding to the original assemble (exchange and addition of components).

The sensors and actuators at this station are similar to the previous ones without the temperature related part and the stirrer. They are also duplicated as there are two containers.

The continuous process to control at this station is the level, measured by an ultrasonic sensor on the second container. To vary the level in the containers it is used the pump. Consequently, to create a faster or slower variation of the level, it is necessary to vary the velocity of the pump. For this purpose it is developed a speed controller explained in section 4.5.2 that is controlled by a 0 to 10 V voltage, generated by the PID (locally) or by an analogue remote signal generated remotely.

The flow sensor is installed with the only purpose of monitoring the system. It has no influence on the control of the process.

On table 4.3 is listed the actuators and sensors from the figure 4.3. The necessary channels for this station are:

- **Analogue signals outputs (AO) -** 0
- **Analogue signals inputs (AI) -** 1
- **Digital signals outputs (DO) -** 5
- **Digital signals inputs (DI) -** 12

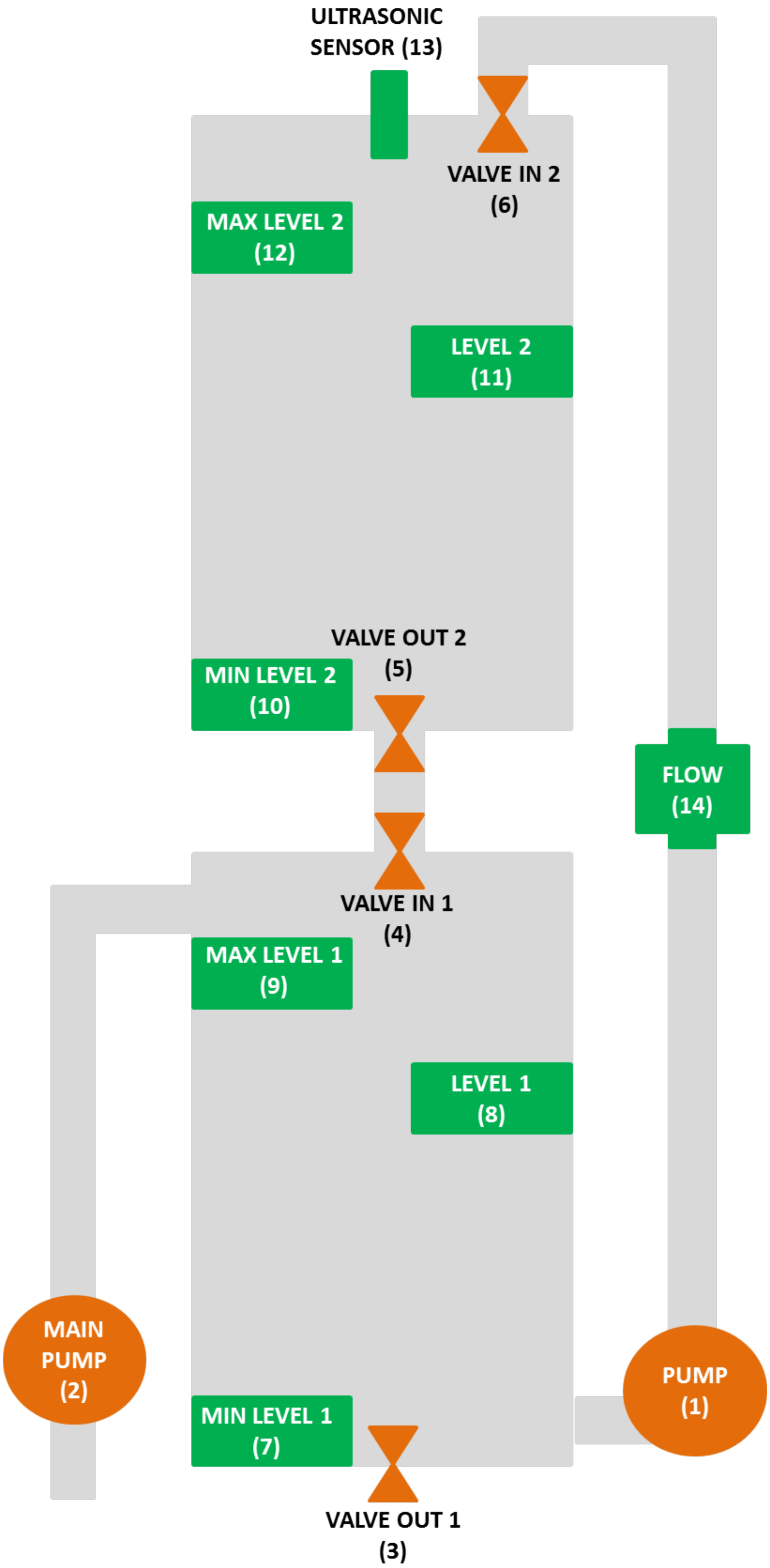


Figure 4.3: Sensors (green) and actuators (orange) available in the Level Station

| Ref. | Variable name | Variable type | Operation Voltage | No. Channels |
|---|---|---|---|---|
| ACTUATORS | | | | |
| 1 | Pump | Analogue | 0 - 10 V | - |
| 2 | Main pump | Digital | 0 / 12 V | 1 DI + 1 DO |
| 3 | Output valve 1 | Digital | 0 / 12 V | 1 DI + 1 DO |
| 4 | Input valve 1 | Digital | 0 / 12 V | 1 DI + 1 DO |
| 5 | Output valve 2 | Digital | 0 / 12 V | 1 DI + 1 DO |
| 6 | Input valve 2 | Digital | 0 / 12 V | 1 DI + 1 DO |
| SENSORS | | | | |
| 7 | Minimum Level 1 | Digital | 0 / 24 V | 1 DI |
| 8 | Level 1 | Digital | 0 / 24 V | 1 DI |
| 9 | Maximum level 1 | Digital | 0 / 24 V | 1 DI |
| 10 | Minimum level 2 | Digital | 0 / 24 V | 1 DI |
| 11 | Level 2 | Digital | 0 / 24 V | 1 DI |
| 12 | Maximum level 2 | Digital | 0 / 24 V | 1 DI |
| 13 | Ultrasonic sensor | Analogue | 0 - 10 V | 1 AI |
| 14 | Flow sensor | Digital (local) | 0 / Vcc V | 1 DI |
| | | Analogue (remote) | 0 - 10 V | - |

Table 4.3: Level station sensors and actuators variables

**Flow station:**

In Figure 4.4 is shown the scheme of the sensors and actuators from the flow automation station with all the alterations regarding to the original assemble (exchange and addition of components).

As the aim of this station is to measure and control the water flow, the only differences from the previous station are the removal of the top container that includes the ultrasonic sensor, and the addition of the proportional valve to control the flow of the water that passes on the pipes by measuring it with the flow sensor.

On table 4.4 is listed the actuators and sensors from the figure 4.4.

| Ref. | Variable name | Variable type | Operation Voltage | No. Channels |
|---|---|---|---|---|
| ACTUATORS | | | | |
| 1 | Pump | Digital | 0 / 12 V | 1 DI + 1 DO |
| 2 | Main pump | Digital | 0 / 12 V | 1 DI + 1 DO |
| 3 | Output valve | Digital | 0 / 12 V | 1 DI + 1 DO |
| 4 | Input valve | Digital | 0 / 12 V | 1 DI + 1 DO |
| 5 | Proportional Valve | Analogue | 0 - 10 V | - |
| SENSORS | | | | |
| 6 | Minimum Level | Digital | 0 / 24 V | 1 DI |
| 7 | Level | Digital | 0 / 24 V | 1 DI |
| 8 | Maximum level | Digital | 0 / 24 V | 1 DI |
| 9 | Flow sensor | Digital (local) | 0 / Vcc V | 1 DI |
| | | Analogue (remote) | 0 - 10 V | - |

Table 4.4: Flow station sensors and actuators variables

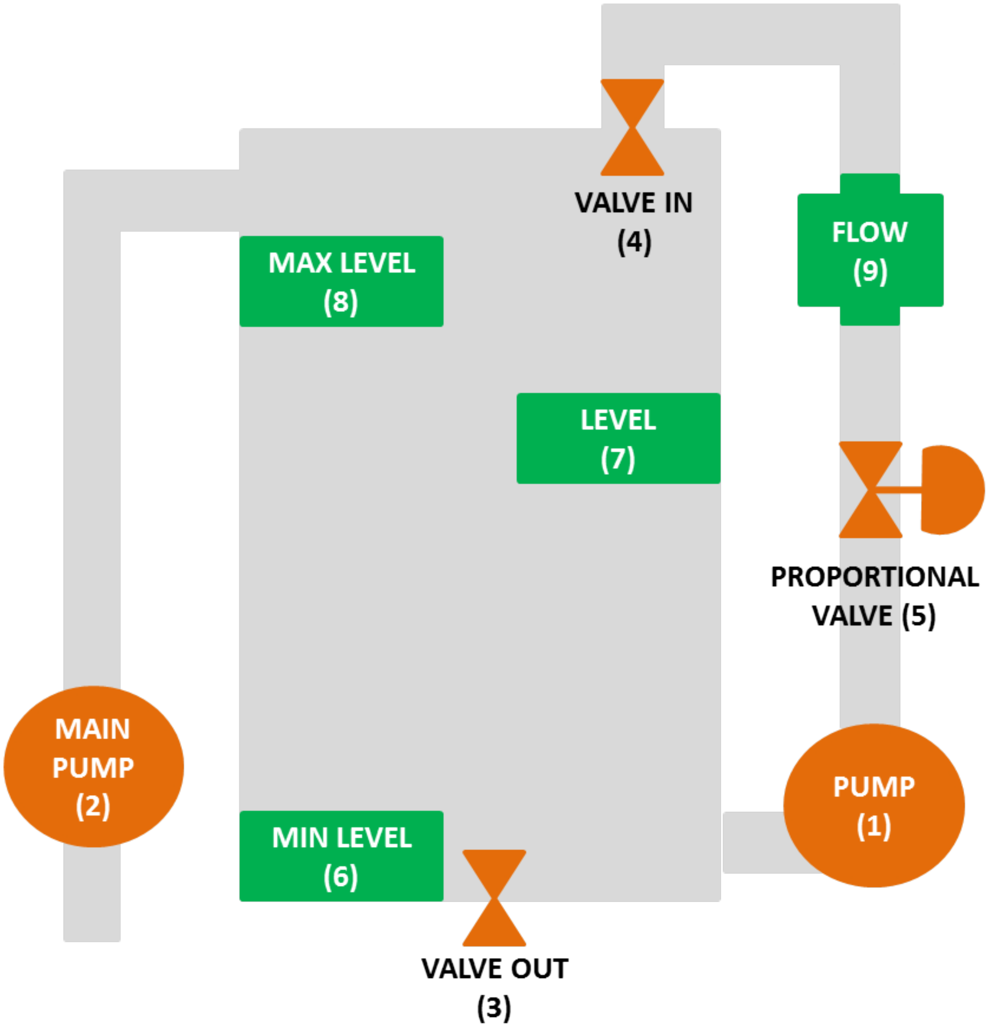


Figure 4.4: Sensors (green) and actuators (orange) available in the Flow Station

In this station, the necessary number of input and output channels are:

- **Analogue signals outputs (AO) -** 0
- **Analogue signals inputs (AI) -** 0
- **Digital signals outputs (DO) -** 4
- **Digital signals inputs (DI) -** 8

Finally it is possible to make the count of all the channels necessary for the project implementation and compare it with the available ones (table 4.5). It is important to keep in mind that there are still some digital output signals that will be required. They are explained on section 4.5.

After this signals analysis and taking into consideration the figure 4.1, it is easy to understand which signals need to be isolated, adapted and controlled.

Starting with the digital signals, it is known, from figure 4.1, that all the signals that are controlled (actuators) or monitored (sensors) by the remote user have to be isolated. Consequently only the two cooler actuators do not need to be isolated, all others require an electrical isolation.

| Signals | No. Available | No. Requested | Balance |
|---|---|---|---|
| **Digital Outputs** | 24 | 18 | 4 |
| **Digital Inputs** | 40 | 30 | 9 |
| **Analogue Inputs** | 4 | 2 | 2 |
| **Analogue Outputs** | 2 | 1 | 1 |

Table 4.5: Comparison between available and requested PLC channels

Secondly it comes the adaptation. As it is shown on table 4.1, the voltages of the PLC digital output channels are different of the actuators operating voltages. Thereby it requires an adaptation of the signal to the desired voltage values of the actuators.

Regarding the analogue signals, once more it is required to isolate all the signals that come in and out from the remote device, which means that all the analogue signals listed earlier need to be isolated.

Finally it comes the control of the analogue signals. As the remote analogue signals do not pass through the PLC, it is necessary to develop a control hardware to select which signal is desired to actuate, the PID (local) or the remote.

Following in this chapter, it is explained each of the subjects referred above.

## 4.4 Electrical Isolation and Adaptation

In many applications of automation in processes control, different voltages and currents are measured from the sensors and actuators in order to monitor and operate them. Often these applications involve environments with hazardous or common-mode voltages, transient signals and fluctuating ground potentials capable of damaging measurement systems and ruining their accuracy. To overcome these issues it is implemented on these automation applications an electrical isolation of the signals. Its aim is to physically separate the original signals source and their ground planes from the rest of the system in order to avoid the damage of the entire system in case of an abnormal voltage variation at the sensors or actuators. The general scheme of an isolation methodology is shown in figure 4.5.

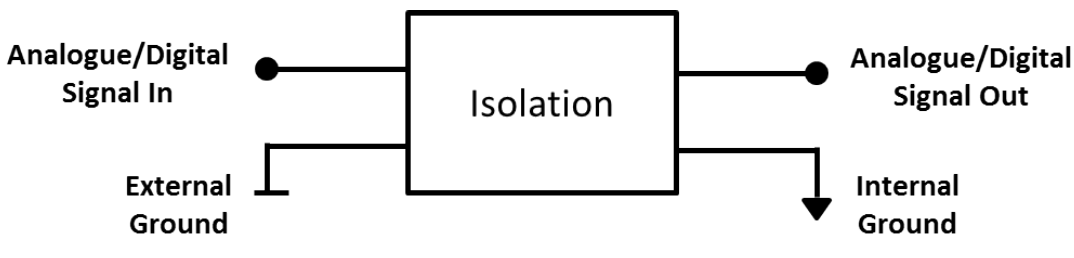


Figure 4.5: Scheme of general electrical signal isolation

For the automation system proposed for this dissertation project, this is an extremely important topic as it is an automation application for didactics purposes. It implies that the system will be used constantly by people with huge lack of experience on the field and it will not be possible to guarantee a constant surveillance by the responsible professors. It is so necessary to create protections for the misuse of this application to avoid eventual damages.

### 4.4.1 Digital Signals

To create a physical separation between the remote signals and the input channels of the PLC it was thought about two options:

1. Electromechanical relays;

2. Optocouplers.

It is going to be briefly described the two options and most important, the advantages and disadvantages of each one for this project. This analysis will help to understand the decision made for the isolation of the remote signals.

**Electomechanical relays:**

The electromechanical relays are often used to control circuits and for isolation purposes. As it is shown on figure 4.6, the relays are made of a coil, an armature mechanism and electrical contacts. When the coil is energized, the induced magnetic field moves the armature that opens or closes the contacts. The left part of the circuit is where the control signal (low voltage) enters, closing the left side circuit that is prepared for a signal of high voltage or current. Therefore it is a very good component to control or isolate circuits with high currents or voltages requirements.

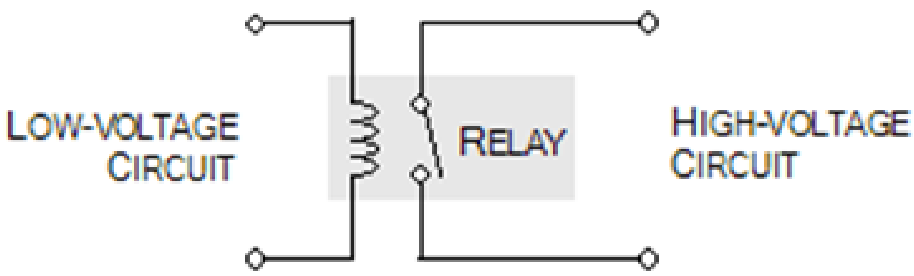


Figure 4.6: Relay internal circuit [2]

As they are mechanical components, there are however some problems that come with it. Firstly, they cannot be switched on and off at high speeds as they have slow response and the switch contacts would rapidly get damaged. Also the coil needs a fairly high current to energize which makes it not useful for micro electronics circuits.

There are also other advantages or disadvantages that are not taking in consideration here, as they are not relevant for this project aim.

To sum up, this component **suits** for this project requirements. There is no need for high speed transactions as the user is a person that will not reach the maximum velocities permitted by the relays. Regarding the current, it is also not a problem, as the remote user device is still not defined and the goal is to install a device that fits to the system requirements.

**Optocouplers:**

An optocoupler is an electric device for signal isolation. It is composed of an optical transmitter, typically a gallium arsenide LED (light-emitting diode) and an optical receiver such as a phototransistor. The two are separated by a transparent barrier which blocks any electrical current flow between the two, but does allow the passage of light. The basic idea is shown on figure 4.7.

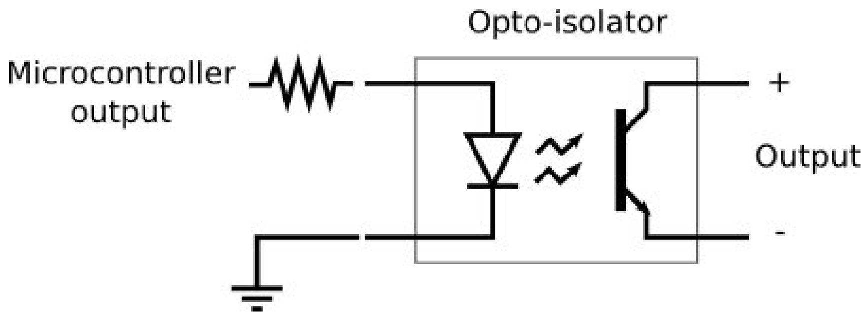


Figure 4.7: Optocoupler electric circuit

The purpose of an optocoupler is exactly the same as the relay. However it surpasses the relay on the switching speed as it is electric and so it depends only on the speed of the LED and the receptor and it is also more reliable. On the other hand the normal optocouplers have low current capacity, which make them not suitable for components that require high current like the pumps of this project.

For the digital isolation it is only necessary to isolate control signals which have a low current, and so the optocoupler **suits** for the isolation requirements. However, when it comes to the signal adaptation, that is done by the same method as it is explained in the next section, it **does not suit** because the pumps have a high current consumption (3.5 A).

After the analysis of these two components it was decided to use relays for the isolation for two reasons:

1. The signal adaptation of the pumps needs a high current and so the relays were the best option. To keep with the same method for all signals it was decided to use the relays;

2. There were already relays available at the laboratory for this project, and as the two options were suitable for the digital isolation, it was decided to use the available material.

**Digital Adaptation:**

For the signal adaptation, as said before, the method used was the same as the isolation. So it was used relays to convert the 24 V signals generated by the PLC to the system requirements voltages and currents.

The circuit schematic is shown on figure 4.8. The diode is used because when it is applied a voltage to a coil it creates a magnetic field. When the voltage is removed, the magnetic field collapses and creates a reverse polarity voltage that can be many times the value of the original applied voltage. This creates a transient voltage pulse that can damage other components in the circuit that are not rated for this polarity or the higher voltage created. Having a reversed biased diode across the coil allows the diode to conduct for reverse polarity voltages and creates a *short circuit* across the coil that allows the pulse to be dissipated in the resistance of the coil wiring.

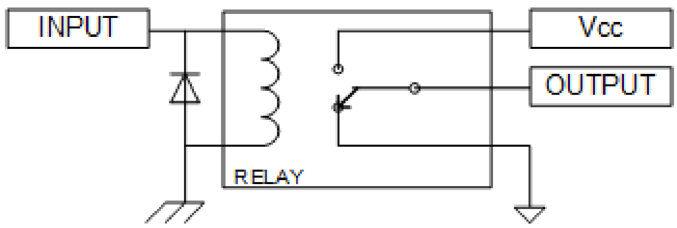


Figure 4.8: Relay circuit used

The only requisite for the relay selection is the maximum switching current, that for the pump should be 3,5 A. For the other components, as the current is low, it can be used relays of 1 A maximum that are already available on the laboratory.

### 4.4.2 Analogue Signals

Isolating analogue signals is slightly harder than digital as the output has to be proportional to the input. The normal procedure for these cases is to use a linear optocoupler. In the laboratory it is available the linear optocoupler *IL300* which, for this reason, is the option for the analogue isolation.

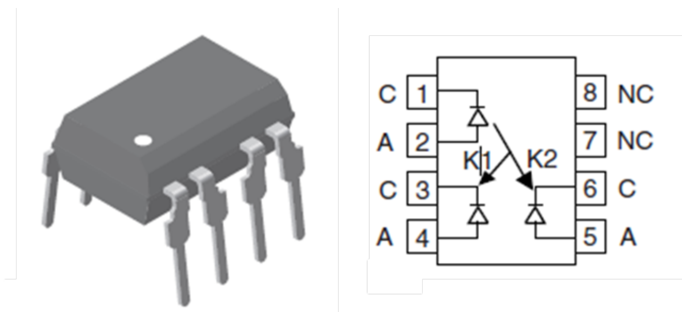


Figure 4.9: *IL300*

The IL300 linear optocoupler consists of an input LED irradiating an isolated feedback and an output PIN photodiode in a bifurcated arrangement. The feedback photodiode captures a percentage of the LEDs flux and generates a control signal that can be used to servo the LED drive current.

This technique compensates for the LEDs non-linear, time, and temperature characteristics. The output PIN photodiode produces an output signal that is linearly related to the servo optical flux created by the LED. The time and temperature stability of the input-output coupler gain is insured by using matched PIN photodiodes that accurately track the output flux of the LED. [**?**]

The configuration of the optocoupler used to have the same output as the input is shown in figure 4.10.

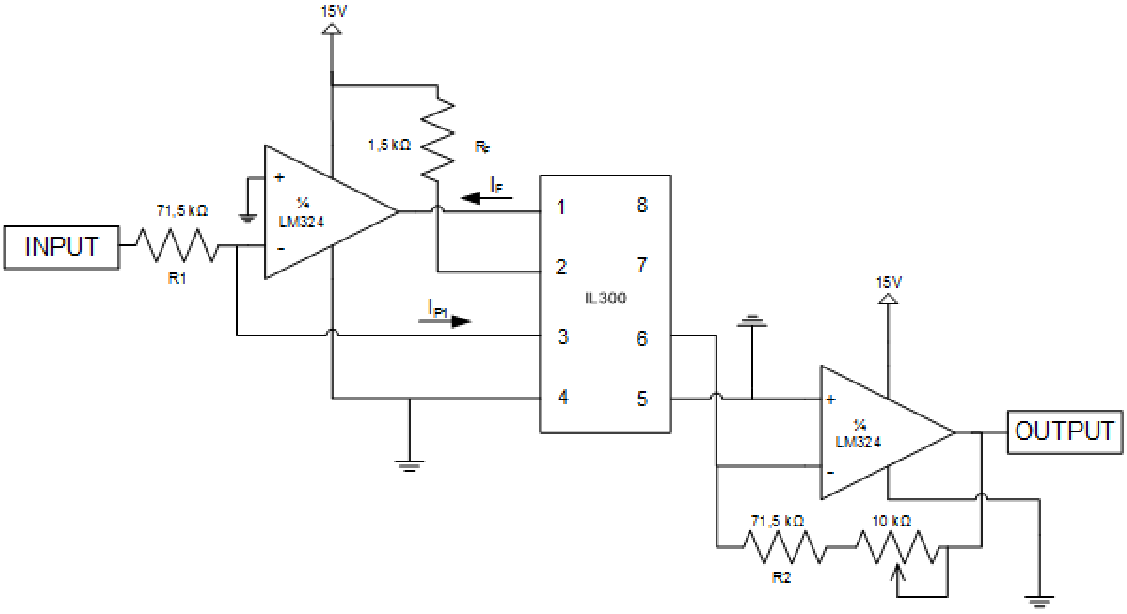


Figure 4.10: *IL300 positive unipolar configuration*

From the datasheet of the *IL300*, the calculations for the dimension of the components are shown below.

$I_F$ is typically $10mA$ and so, with a voltage supply of $15V$, the resistance $R_F$ should be equal to $1,5K\Omega$.

With $I_F = 10mA$, $K1$[1] and $K2$[2] are typically $0,007$ and so:

$I_{P1} = K1 \cdot I_F = 70\mu A$

$R1 = \frac{V_{IN}}{I_{P1}} = \frac{10V}{70\mu A} \cong 148K\Omega$

As K3[3] is typically one and the gain required ($G = V_{IN}/V_{OUT}$) is also one, it is now possible to calculate $R2$:

$R2 = \frac{R1 \cdot G}{K3} = R1 \cong 148K\Omega$

The potentiometer in front of R2 is just for adjustments purposes.

[1]Servo gain: the ratio of the input photodiode current (IP1) to the LED current (IF) i.e., $K1 = I_{P1}/I_F$.
[2]Forward gain: the ratio of the output photodiode current (IP2) to the LED current (IF), i.e., $K2 = I_{P2}/I_F$.
[3]Transfer gain: the transfer gain is the ratio of the forward gain to the servo gain, i.e., $K3 = K2/K1$.

### 4.4.3 Flow signal

The adaptation of the flow signals to both, analogue and digital signals, are respectively explained in this section.

#### 4.4.3.1 Frequency to Voltage Conversion

As said in the beginning of this chapter, the flow sensor generates a square wave variable in frequency and it is necessary to convert this signal to a voltage variance for the remote user.

For this procedure it was chosen the frequency to voltage IC[4] *LM2917N* as it was already available in the laboratory for this purpose.

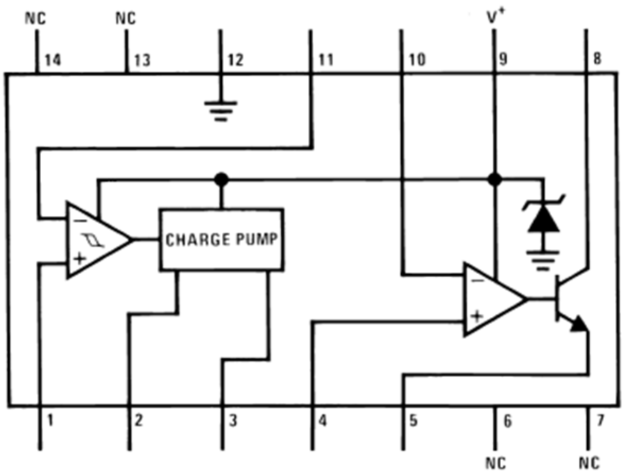


Figure 4.11: *LM2917N* connection diagram [3]

The LM2917 series are monolithic frequency to voltage converters with a high gain op amp comparator designed to operate a relay, lamp, or other load when the input frequency reaches or exceeds a selected rate.

From the datasheet of the component we can dimension the circuit as it is exposed below:

There are some limitations on the choice of R1 and C1 which should be considered for optimum performance. The timing capacitor provides internal compensation for the charge pump and should be kept larger than 500 pF for very accurate operation. Smaller values can cause an error current on R1, especially at low temperatures. Several considerations must be met when choosing R1. The output current at pin 3 is internally fixed and therefore VO/R1 must be less than or equal to this value. If R1 is too large, it can become a significant fraction of the output impedance at pin 3 which degrades linearity.

The size of C2 is dependent only on the amount of ripple voltage allowable and the required response time.

The maximum frequency of the flow sensor is: $25L/min * 752pulses/L \cong 320Hz$ ;

[4]Integrated Circuit

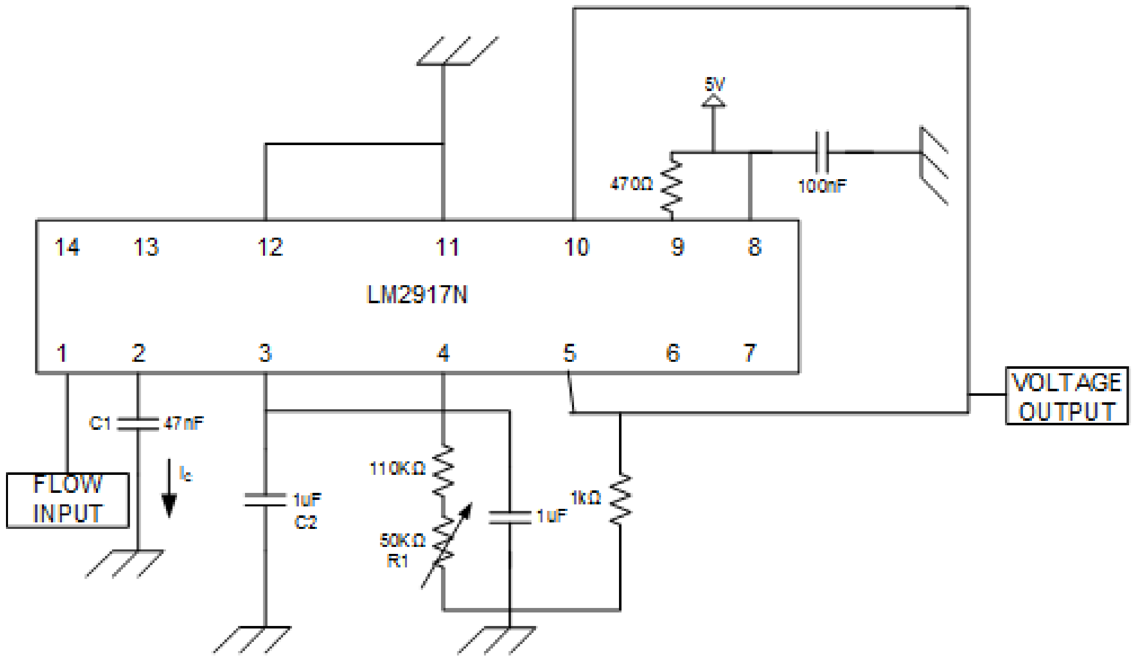


Figure 4.12: *LM2917 frequency to voltage configuration*

Opting for a $C1 = 47\eta F$ and $V_{CC} = 5V$ ;

$I_{C(AVG)} = V_{CC} \cdot f_{IN} \cdot C1 = 5V \cdot 320Hz \cdot 47\eta F = 72,2\mu A$

As we want a variable voltage between 0 - 10 V:

$V_O = i_C \cdot R1 \iff R1 \cong 133K\Omega$

C2 was chosen by experimentation and the rest of the components followed by the datasheet.

#### 4.4.3.2 Voltage Adaptation

The signal generated by the flow sensor is given in a 0,5 V square wave with a variable frequency. As the PLC digital input channels need a 24 V input signal, it is necessary to convert this signal. Normally it would be used a simple relay to make this conversion. However, due to the high frequency of the square wave, it is not possible to use a relay. Also, as the signal will be only connected directly to the PLC, there is no need to isolate it. For these reasons it was used a simple op amp to amplify the signal. The op amp chosen is the *LM324AN* as it is available on the laboratory. It could be used a transistor, however, as there were still some op amps available (the *LM324AN* has 4 op amps each and there were still two op amps available), it was opted for the op amps. The circuit is shown on figure 4.13.

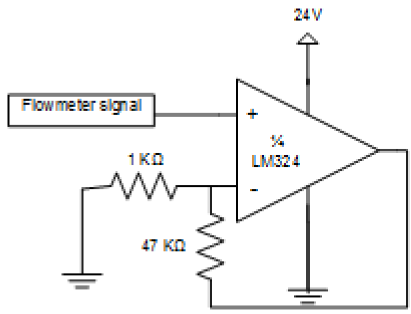


Figure 4.13: Flowmeter voltage adaptation

## 4.5 Control

At this section it is exposed two control parts of the system. The first one is the analogue signals control for the PID and the remote user. This physical system allows the local PLC to select which user is controlling the system, even if the signals are not being generated by the PLC.

The second one is, as referred in the section 4.1, the motors speed controllers.

### 4.5.1 Signals Control

In figure 4.1, and as explained before, due to the lack of analogue channels, the analogue signals need to have a physical control so that the user can opt between the local or remote signals. Consequently it was design a simple control with relays, that uses control signals generated by the PLC to select which device should control the system, the PIDs or the remote signals.

#### 4.5.1.1 Temperature Control

In this station there are two different signals to control: the stirrer and the heating.

The stirrer has two modes of operation - digital and analogue - and two possible controllers - remote or local (PLC). In the PLC there is only the necessity of having one analogue signal, as it can be programmed internally for the desired value. On the other hand, for the remote user is required two different signals, a digital and an analogue, as the aim of the system is to allow the students to control the stirrer in these two different ways.

The designed circuit is shown on figure 4.14.

There are two internal variables from the PLC created to generate the control signals *Stirrer_RL* and *Stirrer_DA*. The modes of operation of the circuit are presented on table 4.6.

| **Signal desired** | **Stirrer_DA** | **Stirrer_RL** |
|---|---|---|
| Digital Signal (Remote) | ON | OFF |
| Analogue Signal (Remote) | OFF | OFF |
| Analogue Signal (PLC) | ND | ON |

Table 4.6: Stirrer operation modes

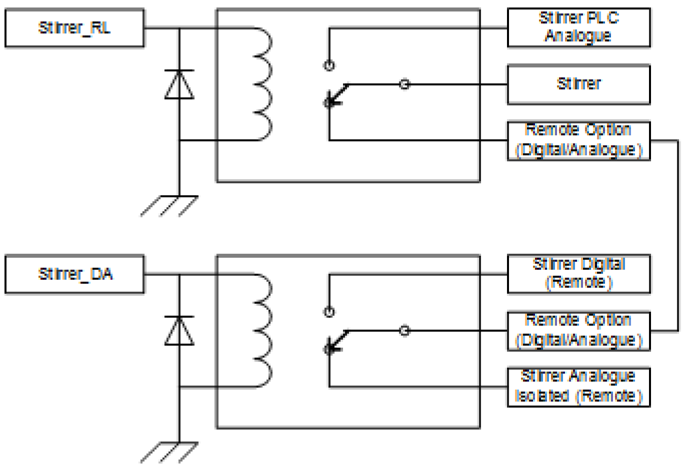


Figure 4.14: Stirrer control circuit

The heating unit has also two modes of operation: analogue and digital. The operation modes are already presented on table 2.1 and the control circuit is shown on figure 4.15.

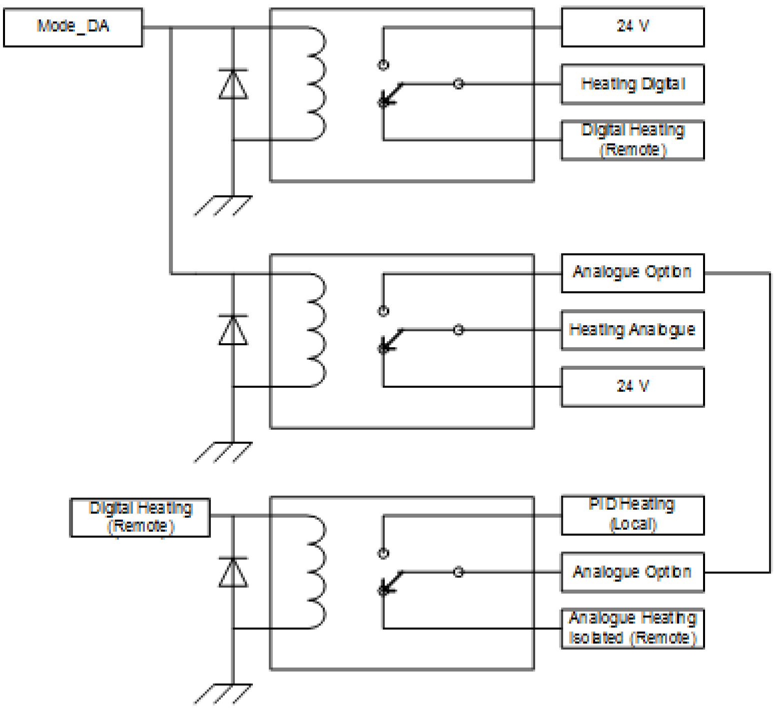


Figure 4.15: Heating unit control circuit

#### 4.5.1.2 Flow and Level Control

The next two stations, flow and level, have exactly the same control circuit as both have only one analogue actuator with one control signal.

As the PID is an external signal, it can not be controlled by the PLC, so it is necessary to guarantee the possibility of a shut-down of the component. For that reason there are two internal variables (*Control 1* and *Control 2*). The first one selects which signal to use and the other turns off the actuator in case of a local request.

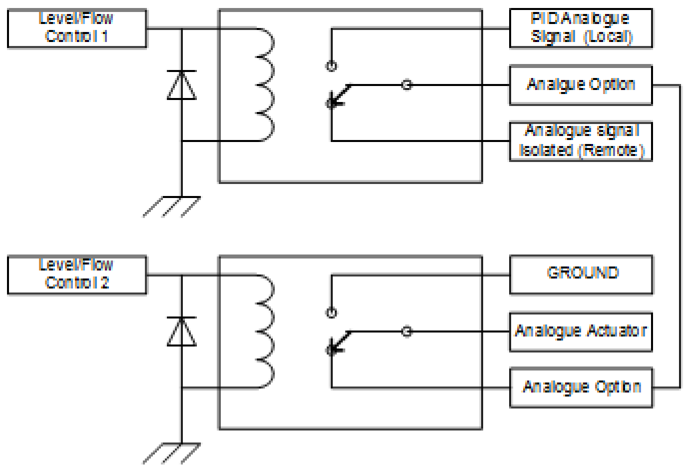


Figure 4.16: Flow and level control circuit

| Signal desired | Control 1 | Control 2 |
|---|---|---|
| Turn OFF the actuator | ND | ON |
| Analogue Signal (Remote) | OFF | OFF |
| Analogue Signal (PLC) | ON | OFF |

Table 4.7: Stirrer operation modes

To make this signals selection independently in each station, it is necessary to use six digital outputs more of the PLC. From the table 4.5 it is possible to see that there are exactly six digital outputs available, and so it is possible to make this implementation. However, in case there was the need of having any extra digital output, it could be used the same control for the level and flow station. It would not allow to make the signals selection independently, but it would fit for this system aim, as the intention is to, or control remotely, or control locally the signals.

### 4.5.2 Motor Speed Control

The last part of hardware to implement is the control circuit of the three motors with speed variation: one stirrer and two pumps. These components have a brushed motor that can be controlled with a PWM[5] controlled by an analogue signal between 0 and 10 V (0 V represent a duty cycle of 100% and 10 V of 0%).

After the generation of the PWM it is also necessary to adapt its voltage and current to the motor requirements. For this purpose it can be used a power transistor with the required voltage and current specifications, to control the motor. As the motor consumes 3,5 A and analysing the components available in the laboratory, it was chosen the *BDX33C* that suits for the specifications required.

For the PWM circuit implementation, considering the components available on the laboratory, there were two options: the IC *NE555* and the IC *TL594*.

Firstly it was decided to use the *NE555* for the simple reason that it was simpler than the *TL594*. The *NE555* is a precision timing circuit capable of producing accurate time delays or oscillation. The function block diagram is shown on figure 4.17 and the implemented circuit on figure 4.18.

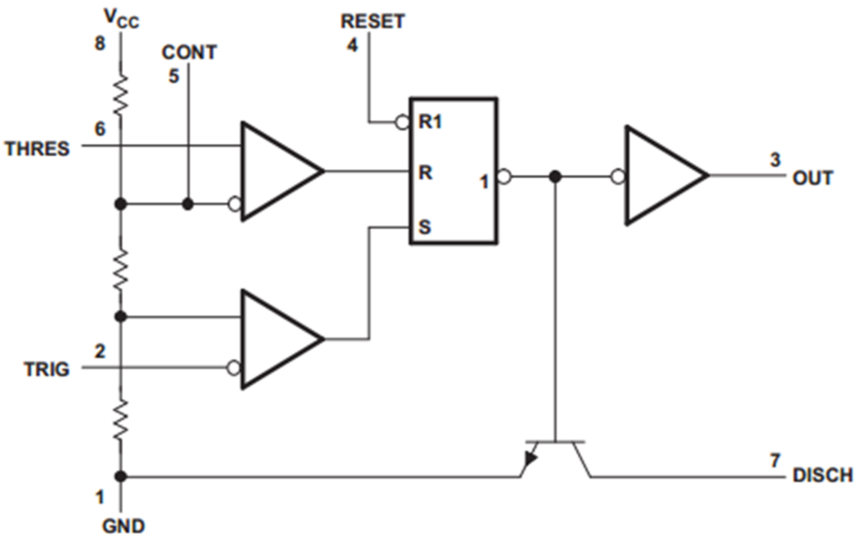


Figure 4.17: NE555 function block

To control the duty cycle with an analogue voltage it was used the *CONT* pin 5 of the IC and the circuit from the datasheet. After the circuit implementation, it was observed that the alteration of the control pin of *NE555* would also alter the PWM frequency which was not pretended.

Consequently it was implemented the controller with the IC *TL594* that is an IC with all the functions required for the construction of a PWM control circuit. Its function block is shown on figure 4.19.

To control the duty cycle of the PWM generated by an analogue input voltage it is used the Dead-time control pin. By the *TL594* datasheet, it is possible to see that this control pin permits operates with voltages between 0 to proximately 3 V, where 0 V corresponds to the maximum duty cycle, and 3 V to zero duty cycle. After testing the circuit, it was observed that the zero duty cycle

[5]Pulse Width Modulation is a modulation technique that generates variable-width pulses to represent the amplitude of an analogue input signal [11]

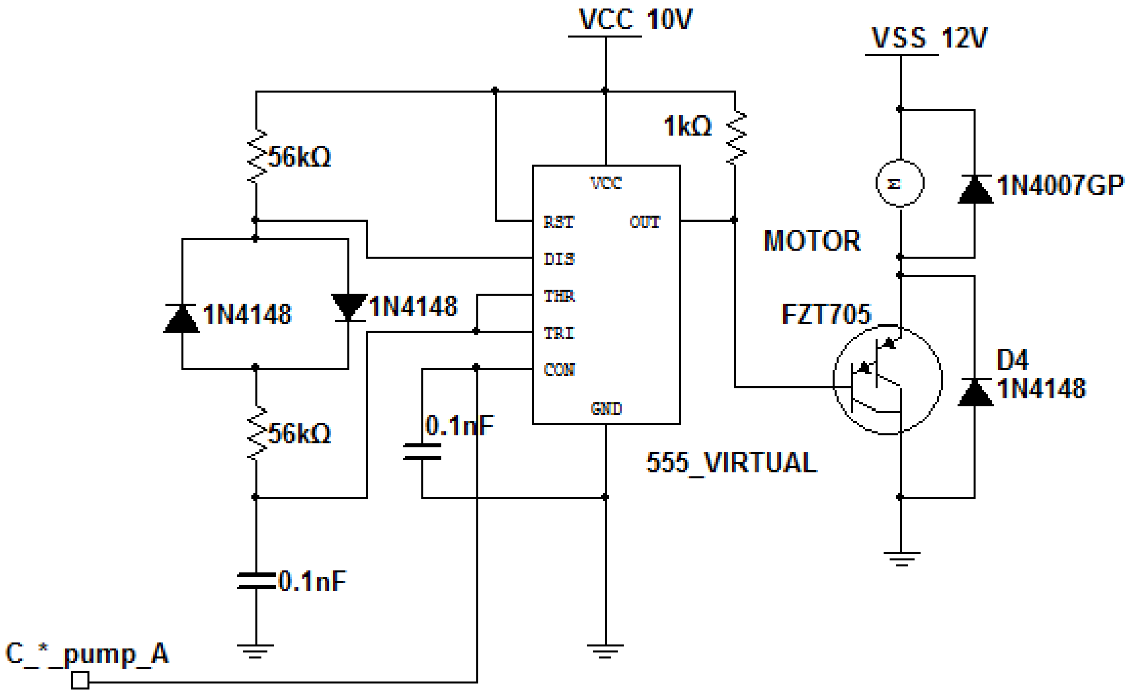


Figure 4.18: Motor speed control circuit with NE555

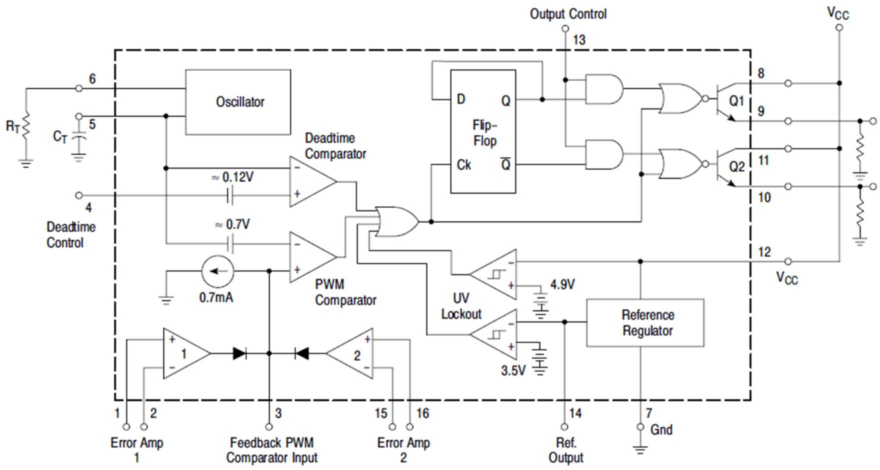


Figure 4.19: TL594 function block

corresponded not to 3 V, but to 2,8 V. For that reason the calculations where made based on the 2,8 V.

As it is desired to control the motor speed with a voltage between 0 and 10 V, it is added a voltage division to reduce the voltage on the DTC (Dead-Time Control) to the correct scale.

After the control signal is adjusted it is necessary to make the 0 V correspond to the minimum speed (zero duty cycle) and the 10 V correspond to the maximum. It is used an inverter *GD74LS02* that is available on the laboratory to invert the signal.

Having the PWM correctly generated, the final step, as it was also planned for the *NE555*, is to supply the motor with the correct output current and voltage. Therefore it is used a *BDX33C* as it is explained before in this section.

The final circuit implemented is shown in figure 4.20.

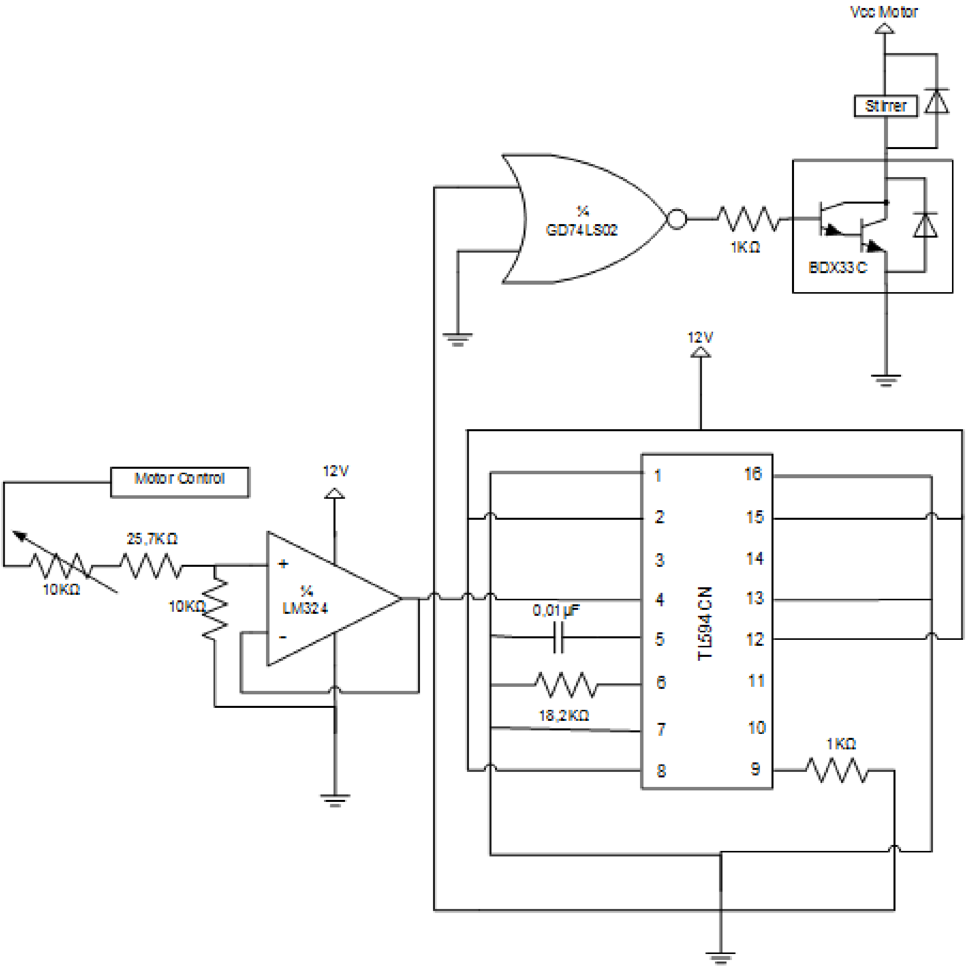


Figure 4.20: Motor speed control circuit with T1594

To determine the capacitor and resistance on pins 5 and 6 it was used the formula below and chosen a frequency of more or less $6K\Omega$:

$$f_{osc} = \frac{1.1}{R_T \cdot C_T}$$

## 4.6 Conclusion

The hardware projection for the automation system exposed in this chapter solves the first part of the problem. With the designed circuits, the system is protected against dangerous physical situations and also, the PLC is prepared to control all the system variable in case of need.

Some of the methods used for this part of the dissertation could be avoided (like the control circuits), if there would be more analogue channels available for the PLC. However, this implementation also suits for this project goals and does not require new equipment for the PLC.

It is also important to notice that there are still some analogue and digital signals available that can be useful for some possible extensions in this automation system. It is also feasible to acquire new channels for the PLC and extend some functionalities of the tank system, like the monitoring of all analogue variables.

# Chapter 5

# Software

In this chapter it is described all the tools needed to develop the user application to control and manipulate the automation system.

Firstly it is explained the PLC proper software - *TwinCAT*. The sections one, two and four present the two different software that are used in this dissertation project for the proper use of the PLC.

It is also described the interlock in section three, that is a software program running in parallel with the control application, used to protect the system against limit situations that can happen due to wrong order to the system.

Finally in the next two sections it is detailed how does the HMI communicate with the PLC and it is presented the final HMI developed for the supervision and control of the system.

## 5.1 Beckhoff TwinCAT 2

To create the control software pretended for the whole system it is necessary to program the PLC with the required instructions. For the HMI is also important that the PLC communicates somehow with an user interface development software, that is going to be explained in this chapter.

For this *Beckhoff* PLC there is already a software available to program it and even to communicate with other computers if desired. The software is called *TwinCAT* and it has three main programs:

1. **TwinCAT PLC Control -** Is a complete development environment for the PLC. Use of the editors and debugging functions is based upon the proven development program environments of advanced programming languages. It permits to link to other programs and computers, even through a network. It is possible thanks to standardized open interfaces (DDE, OCX, DLL, etc). Remote access is also possible with this development environment.

2. **TwinCAT System Manager -** Is the central tool for the configuration of the *TwinCAT System*. The inputs and outputs of the participating software tasks and the physical inputs and

outputs of the connected field buses are managed by the *TwinCAT System Manager*. Additionally the online values of the active configurations can be monitored. The logical inputs and outputs are assigned to the physical ones by logically linking variables of the software tasks and variables of the field buses.

3. **TwinCAT Scope View -** Is an analysis tool providing graphical display of the variables related to various PLC tasks.

The figure 5.1 represents the structure of the *TwinCAT* software. As it is shown, the top part of the figure represents the software to be developed, respectively the HMI, the *TwinCAT system* to link the virtual variables to the physical ones, and the *PLC Control* with the implemented code. After it communicates with the PLC by a TCP/IP protocol or by a COM port, that generates the signals to the physical system as it is shown in lowest part of the figure.

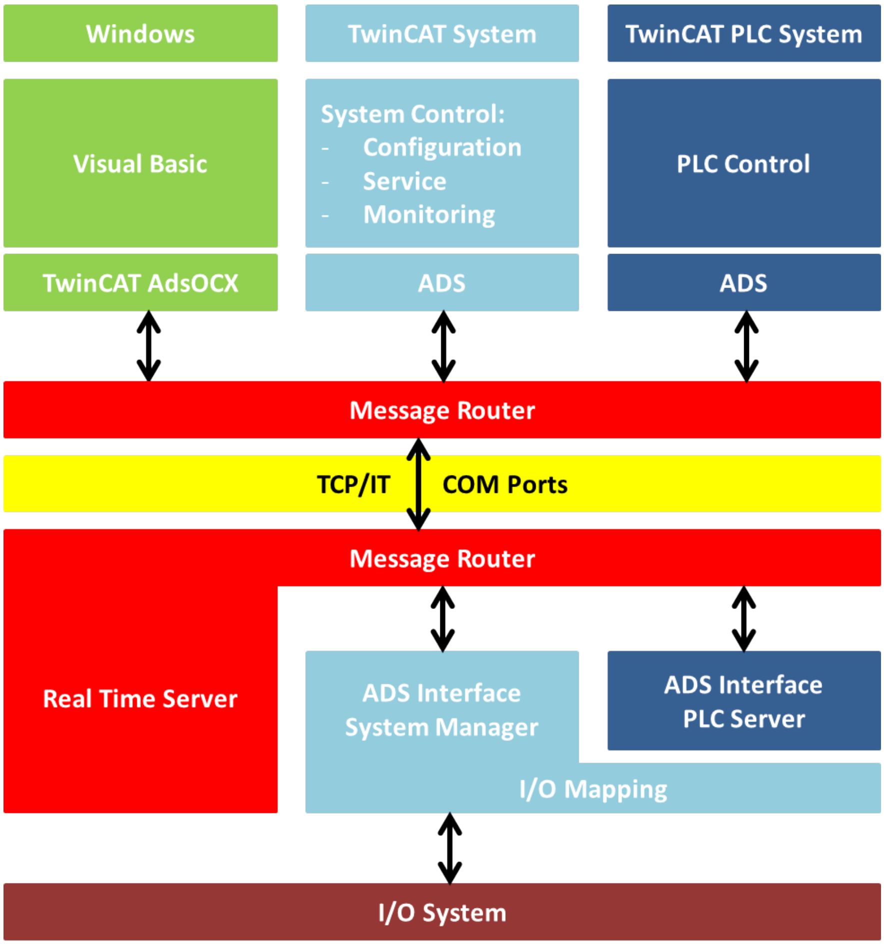


Figure 5.1: TwinCAT software structure [4]

It is now important to understand what to implement and where to implement.

The first important requisite is the control of the system, that means to implement all the possible functions and operations that can be order to the system and respective sensors or actuators necessary to monitor or change. Every time an order comes from the local user it will be executed by the PLC by this software control.

Behind this program, there is an interlock constantly monitoring the process of the system. Every time there is an exceptional event that can damage the system, it acts to repair the failure (turn off or on the necessary actuators). These two programs are programmed in *grafcet* and *ST* language that is accepted by the *TwinCAT PLC Control*.

Finally there is a HMI, developed in *Visual Basic* as the *TwinCAT PLC Control* has an open interface (*OCX*) that can communicate to a remote / local computer with a VB[1] application.

## 5.2 PLC Control

The *PLC Control* software permits the user to program the PLC to run the pretended solution to control the system. This software from *Beckhoff* accepts several programming languages from *IEC 61131-3 standard*[2] to develop the code as [12]:

- ***IL* (Instructed List -)** is a low level language that uses very simple instructions, with some similarities with assembly language programming. It is not commonly used on a daily basis.
- ***LD* (Ladder Diagram -)** is a graphical programming language quite accessible for people familiar with electric systems as it is based on circuits diagrams of relay logic.
- ***FBD* (Function Block Diagram) -** is also a graphical programming language that describes a function between inputs and outputs. It can be internally programmed by other languages.
- ***SFC* (Sequential Function Chart) -** very similar description as the function block diagram. It has steps and transactions the sequentially define the operation of the system by analysing the signals requested each step.
- ***ST* (Structures text)-** is a high level language that is block structured and is normally used with other languages (e.g. *SFS* or *FBD*)

From these programming languages it was chosen *SFC* and *ST* to program the PLC for the only reason that the programmer, in this case the author of the dissertation, had already experience with them.

The figure 5.2 demonstrates the implementation window of PLC Control.

The implementation of the control program of the system, as it is basically communicating the instructions from the local or remote user to the final system, is quite simple. The first and main part of the program is to declare all the variables of the system - remote and local - and allocate the desired ones to a physical memory of the PLC (as it is shown in the *LD Declaration Editor* of figure 5.2). After this implementation, each modification of the variables in the HMI or the remote

[1] Visual Basic

[2] IEC 61131-3 it is a part of the International Electrotechnical Commission standards that specifies the syntax and semantics of a unified suite of programming languages for programmable controllers

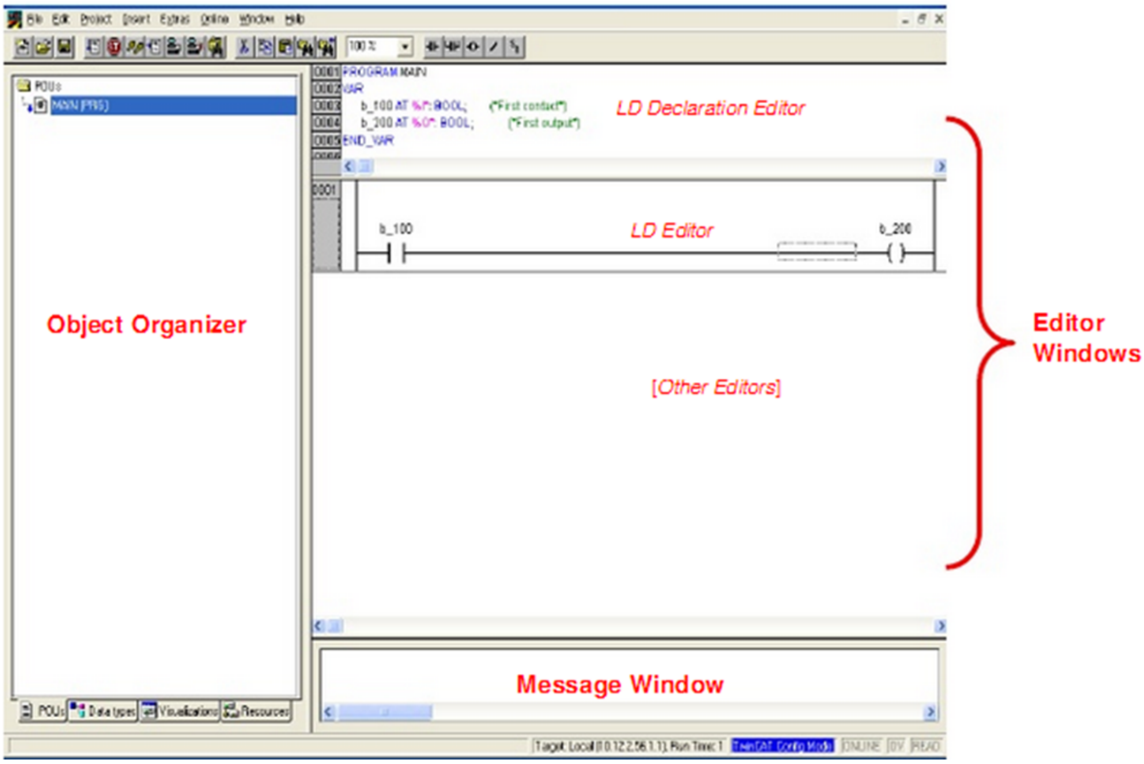


Figure 5.2: PLC Control development environment

user will alter directly the system. There are however some small functions created to facilitate the system control.

**Automatic control for the tanks fill:**

It is important to allow the user to automatically fill the tank system by the HMI. Instead of the manual control of turning on and off the pump, the PLC creates a variable that when activated by the HMI the PLC runs this operation automatically and it permits to choose the level pretended - medium, full or specific (in the case of the level station as it has an ultrasonic level sensor). The states diagram is shown on figure 5.3 and the HMI with the corresponding orders panel is shown in section 5.6.

**Emergency stop:**

As in all automation systems, it is extremely important to have a button of emergency stop. This button when activated by the local user stops all the actuators on the system, even if the system is being controlled by the remote user.

**Control Selection: local or remote:**

Since the beginning of this report it has been discussed the remote or local user control. This option is programmed to be controlled by the local user with the HMI. When the local user chooses the remote control, all the variables in the PLC of the actuators will be copied from the remote device to the final system and the control variables for the analogue signals will be changed to

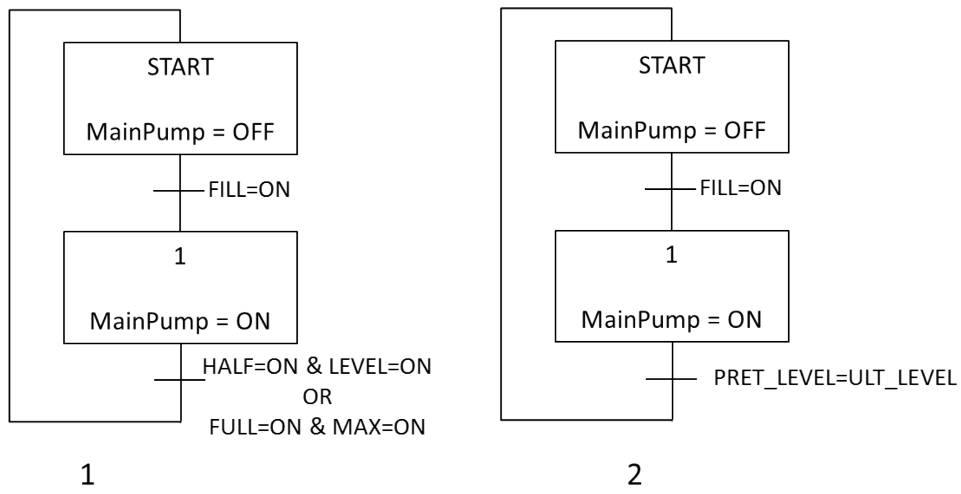


Figure 5.3: States diagram of fill operation: 1) In flow and temperature stations; 2) In level station

activate the remote control, leaving the local control aside. However the emergency button and the user selection will be always priority over the remote control. The analogue variables can be also controlled in parallel with the global system. So, if the local user wants to leave only the analogue variables to be controlled by the remote user, it can be done by just selecting this option.

## 5.3 Interlock

When dealing with an automation system all the cautions are not enough as it is very expensive and a simple error can lead to the system damage. More than an electrical physical protection, it is also crucial to have a monitor program to analyse all actions ordered by the user to avoid danger instructions that can create any kind of risk to the system. In this project it is even more important, as it is a didactic application and it is more probable that inexperienced people like students can give wrong instructions to the system. As an example, if for some reason a person binds a pump until the water overflows, it would probably damage the system. These situations have to be monitored and rectified automatically by the PLC control.

Initially it was thought about creating this protection by hardware. However, as this *Beckhoff* PLC have a simple development programming environment, creating a software program is faster (as it depends only on the software developer), more flexible (as it can be altered very easily) and it was also a preference of the author.

This software program is running independently of all the rest of the system and it is not controlled by any user. Any time the system is operating, the interlock is also running.

The first step for this process is to think about possible risky situations that can happen in the automation system so that they can be avoided. These situations are divided in global situations (they can happen in all stations) and specific situation (they can only happen in a specific station) and are listed on table 5.1.

| REF | Emergencies | Conditions | Actions | Priotity |
|---|---|---|---|---|
| | | **GLOBAL** | | |
| 1 | Water Reaches<br>the maximum level | Maximum_level = True | Input_Valve = ON<br>Output_valve = ON<br>Main_pump = OFF<br>Mixer = OFF | **** |
| 2 | Pressure in Input Valve | Input_Valve = OFF<br>Pump = ON | Pump = OFF | *** |
| 3.1 | Pump working<br>without water | Minimum_level = True<br>Pump = ON | Timer=ON | * |
| 3.2 | | ***WHILE*** *Timer<5s* ***IF***<br>*Minimum_level=False*<br>*OR Pump = OFF* ***THEN*** | - | |
| 3.3 | | ***ELSE*** | Pump = OFF | |
| | | **TEMPERATURE STATION** | | |
| 4 | Temperature<br>reaches 40°C | PT100 = 40°C<br><br>***WHILE*** PT100 > 35°C | Heating_dig = OFF<br>Heating_ana = ON<br>Ventilator_1 = ON<br>Ventilator_2 = ON<br>Main_pump = ON<br>Valve_OUT = ON<br>Valve_IN = ON<br>Pump = ON | *** |
| 5 | Mixer working<br>without water | Level = False<br>Mixer = ON | Mixer = OFF | ** |
| | | **LEVEL STATION** | | |
| 6 | Water reaches the<br>maximum level in<br>tank 1 and 2 | Maximum_level_1 = True<br>Maximum_level_2 = True | Input_Valve_1 = OFF<br>Output_Valve_1 = ON<br>Input_Valve_2 = ON<br>Output_Valve_2 = ON<br>Main_pump = OFF<br>Pump = OFF | ***** |
| | | **FLOW** | | |
| 7 | Pressure in<br>Proportional Valve | Prop_valve = 0%<br>Pump = ON | Prop_valve=100% | *** |

Table 5.1: Interlock rules

**Note:** At the level station the rule number one is replicated for both tanks.

If these situations happen during the normal system operation, the correction actions will replace the previous instructions for that actuators, and a red alert will pop-up in the HMI alerting the user about the situation. After the situation becomes regular, the alarm turns green, informing the user that the problem is solved. The demonstration of these user interactions are show on section 5.6.

Some of the situations described on table 5.1 are not crucial for the normal operation of the system. That is why there are different priorities of the emergency situations. The lowest emergency situations have the only purpose to allow the system to work in better conditions. For that reason, when the low priority situations happen (under *3**) it is asked the user to confirm the interlock actions.

## 5.4 TwinCAT System Manager

In the previous section it is described the general overview of the *Beckhoff* programming interface. Now it is important to expose exactly how it is connected the PLC to the system through the software.

After having all the programs developed to control the PLC it is necessary to create the bridge between the device and the final system. To make it possible the *TwinCAT System Manager* allows to connect each variable created on *PLC Control* to the correspondent physical actuators or sensors.

The first thing to do is to make this software communicate with the PLC. To make it possible, there are two options:

1. **PLC CPU -** If the developed programs are running in the PLC CPU, the *System Manager* will automatically recognize the PLC by its local ID. All the software necessary is running at the PLC CPU on this option.

2. **Remote CPU -** It is also possible to communicate by a remote CPU. In this case, the computer has to be connected in the same local network and then configured by introducing the NetId (Local ID of the PLC) on the configurations options.

Initially it was though about running the application directly in the PLC CPU and controlling it by a console available for the PLC. However, due to the lack of space on the CPU hard disk, this option was not possible. Therefore the application and programs were developed and tested by a remote computer. Lately, on section 5.6 it is explained the solution, between remote or local CPU, for the final system.

After the PLC is communicating with this software, the steps to finalize its configuration are shown in figure 5.4 and described below:

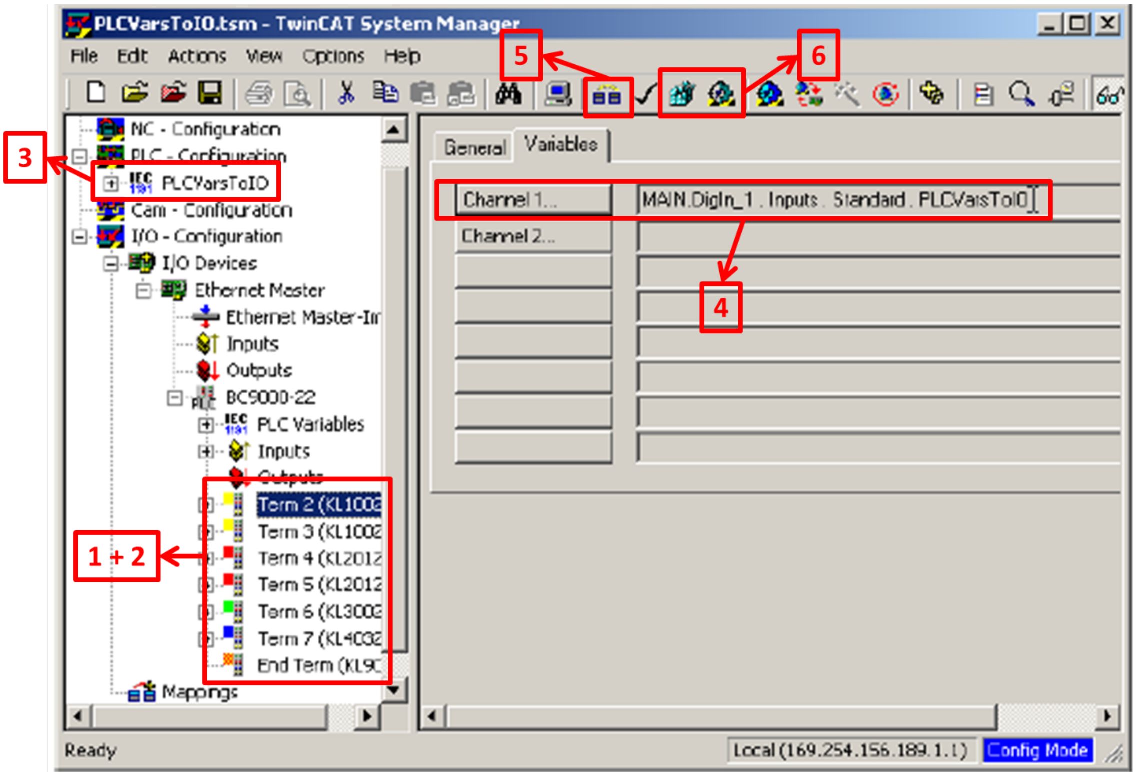


Figure 5.4: System Manager configuration environment

1. Configure the software to recognize the PLC device in order to communicate with it;
2. The system manager will recognize all the channels connected to the PLC CPU;
3. Append the *PLC Control* project to the *System Manager* in order to obtain all variables available in the program;
4. Allocate the desired programmed variables to the physical channels available;
5. Map all the variables network;
6. Build and run both projects, the one from *PLC Control* and the one from *System Manager*.

After running the program, this software also allows the user to monitor each allocated variable of the system in real time as it is shown in figure 5.5. It is not so complete as the *Scope View* but for testing purposes is ideal.

With the PLC configurations done, the application is ready to interact with the actuators and sensors of the final automation system.

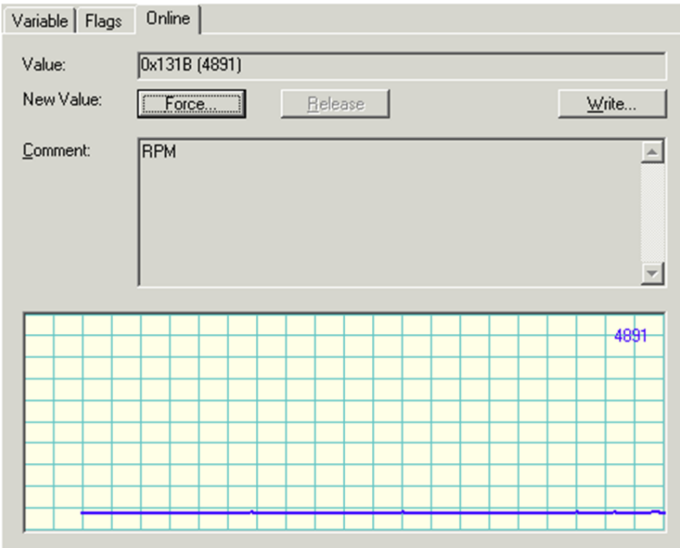


Figure 5.5: System Manager online tab

## 5.5 Visual Basic Communication with TwinCAT

Before implement the HMI application by *Visual Basic*, it is necessary to understand how does the *TwinCAT* software communicates with an external device. This PLC uses the Automation Device Specification (ADS) to exchange messages between different and independent devices. In VB there are different libraries that can be used to communicate with the PLC. For this HMI, as it is shown on figure 5.1, it is used the library called *TwinCAT AdsOCX* for the only reason that it was the preferred of the author. Other possible library could be *ADS.Net*.

To follow with the implementation it is first described the necessary functions of this library and how to use them.

The interest on this project is to read and write different type of variables on the PLC from the HMI that is created in VB. The *AdsOCX* library permits to connect the appended variables from *System Manager* to virtual variables created on VB. There are three different ways to create this connections: [13]

1. **Synchronous -** In this operation mode, once the functions are called, the VB program is interrupted and it only continues after the requested data is available. It is ideal for programs that are connected directly to the PLC CPU as the connection is very fast and also in programs that require the user to introduce different values and submit them all together. In this dissertation application it is used to alter the actuators values, as it is essential to guarantee that the application alters the actuators values before proceed.

2. **Asynchronous -** In this case, the VB program is not interrupted after the function is called. The application operates normally until the requested data arrives as a parameter to the VB program. It is ideal when the PLC and the VB program are in different locations as the communication speed can be slower.

3. **Connect -** This operation mode creates a direct connection between a variable from the *System Manager* and a variable from the VB program. Every time there is an alteration of the variables, in VB or physically, an event is called to transfer the referred data to the respective destination. It is so ideal for this application aim on the sensors monitoring. Any time a sensor alters, it is communicated to the HMI, altering the virtual variables.

As said, in the application developed, it was used the *Connect* operation mode to read and update the sensors, and the *Synchronous* operation mode to change the actuators after the user orders. It is so briefly described these functions below.

- **AdsReadVarConnectEx -** Creates a fixed connection between a VB variable and a *System Manager* variable. Every time there is an alteration the event *AdsReadConnectUpdateEx* is called and different orders can be made (e.g. updating the sensors states on the HMI):

object.AdsReadVarConnectEx(
*adsVarName* As String, **(1)**
*nRefreshType* As ADSOCXTRANSMODE, **(2)**
*nCycleTime* As Long, **(3)**
*phConnect* As Long, **(4)**
*hUser* As Variant, **(5)**
As Long

**(1)** Name of the *System Manager* variable
**(2)** Type of data exchange between VB variable and *System Manager* variable
**(3)** Read cycle in ms
**(4)** Contains an unique handle for the connection that has been established to VB
**(5)** Optional: This value is passed when the AdsReadConnectUpdateEx() event is called

To write on the variables, as said before, it is used the synchronous mode. The write function of this mode requires first to create a handle on VB that will be related to the variable of the *System Manager*. The function that permits this functionality is:

- **AdsCreateVarHandle -** Generates a unique handle for a *System Manager* variable:

object.AdsCreateVarHandle(
*varName* As String, **(6)**
*hVar* As Long) **(7)**
As Long

**(6)** Name of the *System Manager* variable
**(7)** Handle of the *System Manager* variable

Finally it is possible to change the actuators by:

- **AdsSyncWriteReq -** Alters and writes data of any type on the *System Manager*:

object.AdsSyncWriteReq(
*nIndexGroup* As Long, **(8)**
*nIndexOffset* As Long, **(9)**
*cbLength* As Long, **(10)**
*pData* As YY, **(11)**
As Long

**(8)** Index group of the *System Manager* variable
**(9)** Index offset of the *System Manager* variable
**(10)** Length of the data in bytes
**(11)** VB variable from which the data is written into the *System Manager* variable

An important aspect from these functions is that all of them return a value that indicates if there was an error or if the connection was successfully made. This value is always checked to inform the user of the connections states and it appears on the HMI the message with the relevant information.

When closing the application is also important to close the fixed connections between the variables and delete the variables handles created. For this purpose it is used the functions:

- *AdsDisconnectEx( 'Handle of the connection between the VB variable and the System Manager variable' )*;
- *AdsDeleteVarHandle( 'Handle of the System Manager variable' ).*

Now that all the theoretical topics about the *Beckhoff* development environment are explained, it is time to expose the final application and its functionalities.

## 5.6 HMI

A graphic interface permits the user to interact intuitively with the system. It uses images, boxes, check-lists and other user friendly components to permit any person to understand the concepts of the system. With these graphical components the user can monitor and order the physical devices by a digital simulation of the system as it is demonstrated later on this section. It is also relevant to know that the HMI in this project is represented in real time, which means that the sensors change in the application just after they physically change in the system, as the actuators change in the system just after the order is made in the HMI.

To create this HMI application it was opted for the *Visual Basic* development environment. The main reason is that, as said and explained in the previous section, the *TwinCat* software has already proper libraries to communicate with VB. It was also important for the decision the fact the *Visual Basic* is one of the most used programming environments for this kind of applications, which facilitates its alteration in case of need. It is also free for *Microsoft* registered users and all the students in FEUP have free licences for *Microsoft* operating systems. In the future any student or professor can edit and alter the application if desired.

### 5.6.1 Application run device

In section 5.4 it is introduced the options between running the application in a remote CPU or in the local PLC CPU. During the all project the application was developed and tested by a personal computer communicating remotely with the PLC. However the initial plan was to run the application in the PLC CPU and use the touch screen shown in figure 5.6 to interact with the system. When tested the application in the PLC CPU it was detected that there was not enough space on the hard disk of the PLC CPU to run the VB application. The current VB development environments (*Visual Studio 2008 or more*) can only build projects with *.NET framework*[3] versions newer than the one installed in the PLC CPU. These newer framework versions require more space than it is available on the hard disk, which make it impossible to run the VB application developed. Therefore there are different possible solutions for this problem:

1. **Increase the hard disk space of the PLC -** Require to buy a new CPU as it is difficult do find additional memory for this PLC. It is important to notice that the PLC was bought more than five years ago;

2. **Create a VB application from the oldest *.NET framework* version -** It would require to have the *Visual Studio 6* that is not available to download on the *Microsoft* website as it is no longer the official software;

3. **Run the application remotely in a different CPU -** Requires other CPU('s) for the system to run the application remotely.

From the options enumerated the only one that is completely excluded is the second one as it is not reasonable to try to develop an application in a software that is extremely hard to find and not even currently used. Later it can be desired to change the application and so it is important to leave this possibility opened.

Having decided that the application would be developed in any *.NET framework* version, the other options can be opened to decide later, as the application can run in both cases and it only depends on the professor or faculty available resources or opinion. However the option chose for now it was to use an available CPU on the laboratory to run the application and communicate remotely with the PLC. There are two computers on the laboratory that can be used for this purpose: one that is incorporated with a touch screen or one without screen and connect the screen on figure 5.6 to it. It is also important to notice that the CPU **has to have the operating system *Windows***

[3]The .NET Framework is Microsoft's comprehensive and consistent programming model for building applications that have visually stunning user experiences, seamless and secure communication, and the ability to model a range of business processes. [14]

***XP* installed**, as it is a requisite for the *TwinCat 2* software. This limitation makes it impossible to install the software in all the computers of the laboratory, as they do not have this operating system installed.

Figure 5.6: Console Panel Beckhoff

### 5.6.2 HIM design

With all the software studied it is time for the HMI application development explanation. The first thing to define is its structure. A good HMI should be easy to understand and easy to manipulate. In order to make it, the structure shown on figure 5.7 was first defined for each station of the project.

As it is shown, the interface is split in five different parts:

- **Station Selection -** As there are three different stations, it is necessary to allow the user to commute between them and choose which one to control;
- **Tank System Figure -** In this part it is represented the general overview of the tank system from the respective station. It also permits the user to monitor the sensors and actuators (that change their colour according to its state) and alter the actuators or other additional functionalities of the station;
- **Control Panel -** There are few options that the user can make besides changing the actuators. These options appear in the panel control;
- **Analogue Signals Graph -** It is very important for didactic purposes that the students can see the evolution of the analogue signals that they are controlling. For this reason it is available a graph for the analogue controlled variables showing the real-time value of the variables;

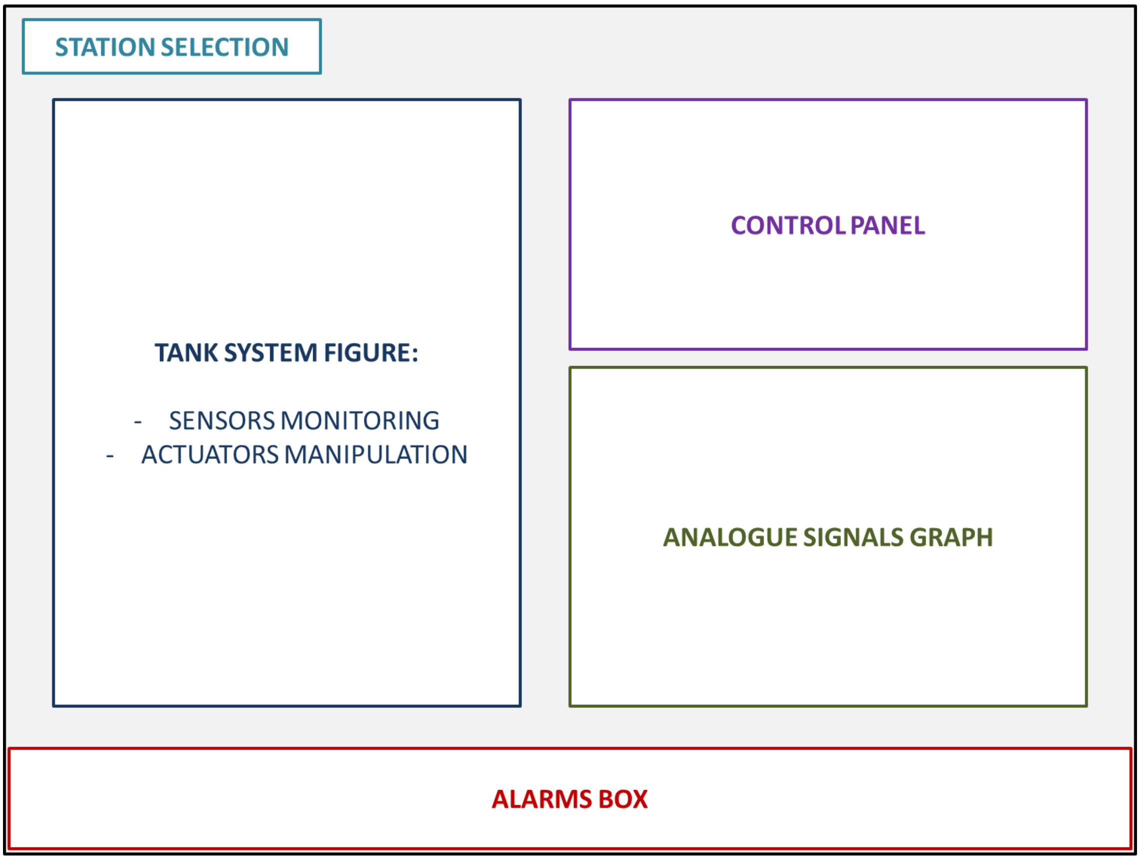


Figure 5.7: User interface structure

- **Alarm Box -** Every time there are emergency alarms or notes to be informed to the user, it can be displayed on the alarms box. Their state will also appear (e.g. if there is an emergency it appears in red, if it is already corrected, it turns green).

The final application design of the temperature station is shown in figure 5.8.

The other stations windows can be seen on the appendix A as they are similar to the figure 5.8 and have almost the same functionalities.

Before the application starts, it is verified if all the variables are communicating with the *System Manager*, and if not, it appears notices saying which error occurred and which variable was. If all the connections are correct, it also appears a notice saying that.

It is easy to understand that all the actuators have red background colour and by clicking on them they will change to green and turn on the respective physical actuators on the system. This option is only possible if the local mode is selected on the control panel. It is also important to notice that if the analogue mode is selected, it appears a bar to choose the intensity of the analogue signal and the digital control stops working.

The sensors are measured and monitored by the water level in the blue bar. If all the level sensors are on, the blue bar will be full; if only the minimum water level sensor is on, the blue bar

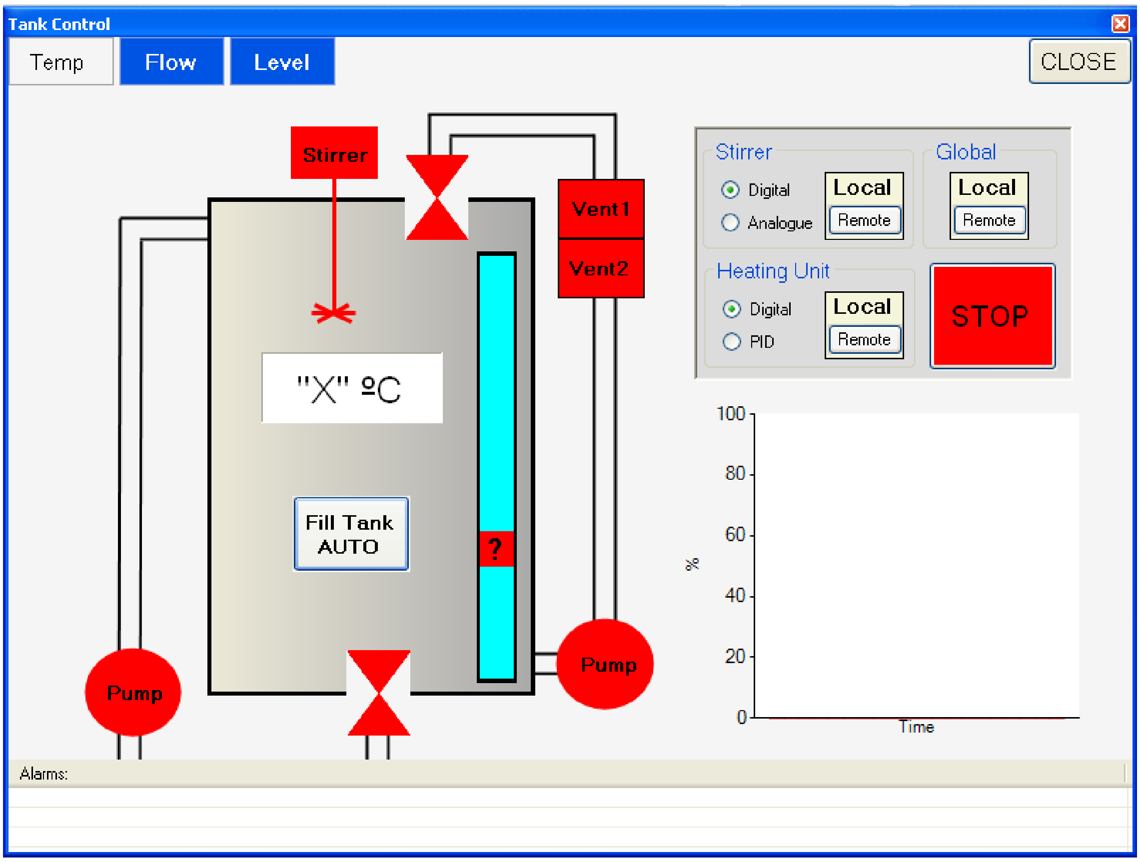


Figure 5.8: User interface final design of temperature station

turns white (without water) and if the minimum and medium water level sensors are on, the blue bar turns half white.

As it is seen the interface is very easy to understand which makes it possible to control by any person without knowing any code behind the application.

## 5.7 Conclusion

In this chapter it is described all the functionalities of the *TwinCAT* software and the development tools necessary for the implementation of the HMI. The final goal is to understand how does the HMI works and explain all the information that is necessary to implement it in order to permit any person to alter the application if desired.

As it is exposed, the HMI is very intuitive and it suits very well for the final purpose of this project: laboratory classes. With the application it is possible to have a total control of the system as well as a constant supervision. It is also protected for bad usages by the interlock, which gives a guarantee to the supervisor professor that the system is well protected and is suitable for students usage.

# Chapter 6

# Didactic Experiments

The last chapters describe the development of the automation system and its functionalities. It is now important to explain how it can be used for didactic purposes.

It is known that this automation system can control continuous variables - temperature, flow and level - by measuring, with sensors, some characteristics of the system. The final purpose of the system, is to allow the students to be able to solve automation tasks for the system control. It is so essential to have the most comprehensive information possible with regard to the static and dynamic characteristics of the processes to be automated.

In this chapter is explained how the students, with the access to this processes characteristics referred in the previous chapters, can control the final system. It means that, by evaluating the behaviour of the variables sensors, they have to be able to estimate the correct values for the respective actuators, in order to maintain the continuous variables on the pretended values. This processes have the possibility to have external interferences installed on the system. Following in this chapter, it is enumerated and explained some tasks that can be done by the students for each individual stations.

To understand the proposed work to the students, it is important to recapitulate the general concept of the closed loop control.

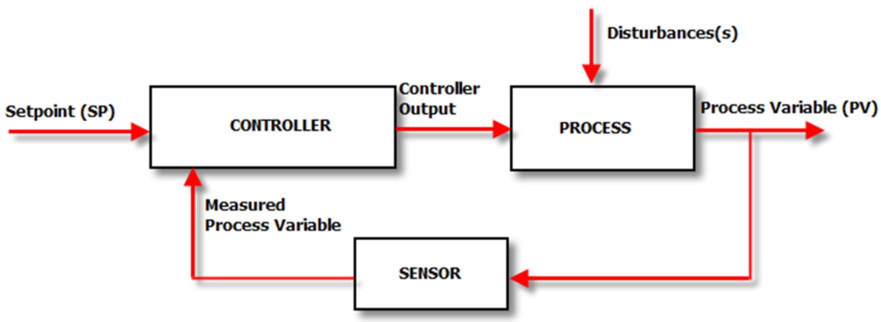


Figure 6.1: Closed loop control diagram [5]

On figure 6.1 it is shown three different aspects:

- **Process -** Corresponds to the physical system to be controlled. This system is explained in the first chapters of this dissertation report;

- **Sensors -** Corresponds to the values of the continuous variables measured by the physical sensors of the system;

- **Controller -** Is an additional system that calculates the values of the output signals for the system stabilization (e.g. maintain the temperature in 30°C). In this system it is installed a *PID* controller, connected to the analogue actuators of the system.

The didactic interest of this system, is to understand and recreate the controller part of the closed loop control. With all the system functionalities available, it is possible to make a list of the main feasible tasks regarding this topic.

Firstly it is important to recapitulate some important notes. There are two modes of operation to control the system, with different functionalities:

- **Local -** All the signals are already connected to the PLC and it has already an independent controller installed, a *PID*. The only way to control the analogue variables is to alter the *PID* parameters. In addition, all the digital variables can be controlled by the HMI.

- **Remote -** The signals are available to be connected to any device and already normalized and isolated. The digital signals are normalized with 24 V and the analogue with 0 to 10 V. There is no control of the signals and it should be developed by the students.

Having this remarks in consideration, there are different tasks that can be done by the students to understand and train the topics of continuous processes control. It is enumerated some of the main tasks and then described them regarding to the available stations.

1. Calculate the *PID* parameters in order to correctly control the analogue variables. This task can be made by configuring the local *PID*, or by creating a new control model in Matlab or other simulation software in order to connect it to the remote user output;

2. Create a new closed loop control model for each station to stabilize the continuous processes of the system;

As each station controls one continuous process, all of them have the same control method. The only aspect that changes is the control model of the stations. Since for the first task the transfer function of the system is not required, it is only explained the possible tasks for the general case, that serves as an example to each station individually. For the second task, as it already involves some specific characteristics of each station, it is specified and detailed the experiment for each station.

## 6.1 Determine the PID parameters

With the *PID* available on the automation system, it is asked to the students to determine the parameters to stabilize the continuous process, in this case, the temperature, level or flow.

Firstly is important to make a brief explanation about the *PID* operation and each parameter.

The *PID* controller is a simple three term controller: [15]

- **Proportional (P) -** Gives a change in the input (manipulated variable) directly proportional to the control error;
- **Integral (I) -** Gives a change in the input proportional to the integrated error;
- **Derivative (D) -** Used in some cases to speed up the response or to stabilize the system and it gives a change in the input proportional to the derivative of the controlled variable.

To adjust the *PID* controller it is required to determine the constant for each of the parameters referred above. They are respectively: *Kr*, *Tn* and *Tv*.

After having these parameters defined, the PID sums the three components and sends the final output to the actuators as it is shown in figure 6.2.

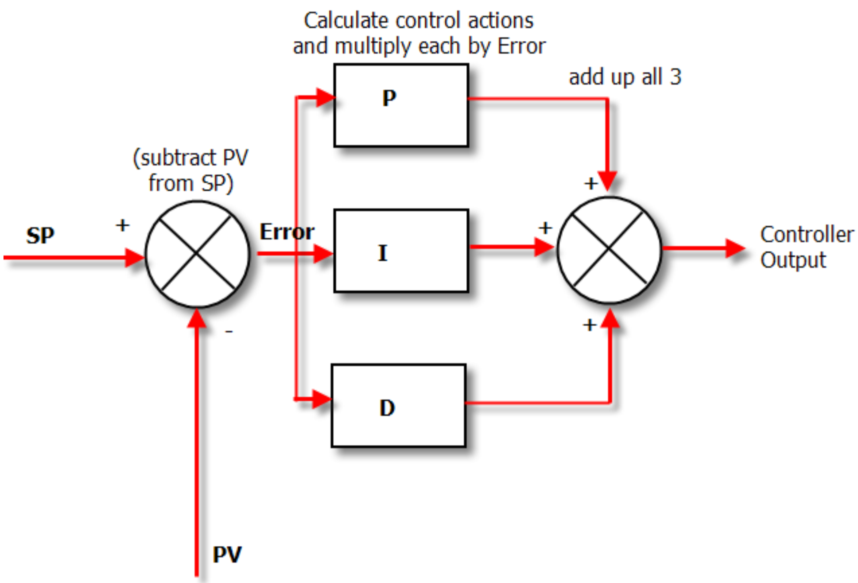


Figure 6.2: *PID* diagram [5]

To correctly adjust the *PID* terms, it is necessary to analyse four major characteristics of the closed loop step response:

- **Rise time -** The time that it takes for the system output to rise beyond 90% of the desired level for the first time;
- **Overshoot -** The difference between the peak level and the steady state;

- **Settling time -** The time it takes for the system to converge to its steady state;
- **Steady state error -** The difference between the steady state output and the desired output.

Considering these characteristics, the students have to optimise the response by determining the three *PID* parameters. For this purpose, it should be considered the figure 6.3.

| Response | Rise Time | Overshoot | Settling Time | S-S Error |
|---|---|---|---|---|
| $K_P$ | Decrease | Increase | NT | Decrease |
| $K_I$ | Decrease | Increase | Increase | Eliminate |
| $K_D$ | NT | Decrease | Decrease | NT |

Figure 6.3: Effects of increasing a parameter independently [6]

Finding these parameters can be hard when there is no references for the specific system. There are however some methods to easily find these parameters. In this section it is explained a simple method that can be used by the students to obtain rapidly a solution for the system stability, even if not the most optimized.

J.G. Ziegler and N.B. Nichols have specified setting rules, which are still widely used today. This rules are ideal for cases like the stations installed in this system, where there is no model available or the closed loop can be operated safely along the stability limit.

This method is composed of five steps: [16]

1. Determine if the *Kr* is positive. To do so, increase the input a little and observe if the resulting steady state value also increased. If so, it means that the *Kr* is positive;
2. Set the controller as a P-controller (*Tv*=0 and *Tn*=0);
3. Turn the controller gain, *Kr*, up slowly (more positive if *Kr* was decided to be so in step one, otherwise more negative if *Kr* was found to be negative) and observe the output response. Note that this requires changing *Kr* in step increments and waiting for a steady state in the output, before another change in *Kr* is implemented;
4. When a value of *Kr* results in a sustained periodic oscillation in the output (or close to it), mark this critical value of *Kr* as *Ku*, the ultimate gain. Also, measure the period of oscillation, *Tu*, referred to as the ultimate period;
5. Based on the two values (*Ku* and *Tu*), the other controller parameters are then to be calculated depending on which controller type is desired, according to the table 6.1.

The values acquired after this method implementation are usually not optimized, but sufficiently good for a workable closed loop control behaviour. However, the students should change manually the parameters in order to try to obtain better results and understand the operation of the controller.

| | ***Kr*** | ***Tn*** | ***Tv*** |
|---|---|---|---|
| P-controller | $0,5 \cdot Ku$ | - | - |
| PI-controller | $0,45 \cdot Ku$ | $0,85 \cdot Tu$ | - |
| PID-controller | $0,6 \cdot Ku$ | $0,5 \cdot Tu$ | $0,12 \cdot Tu$ |

Table 6.1: Ziegler/Nichols rules [8]

Other methods could be used to determine the *PID* parameters. However this method was chosen as it is widely used today.

The figure 6.4 summarizes the steps for the proposed task.

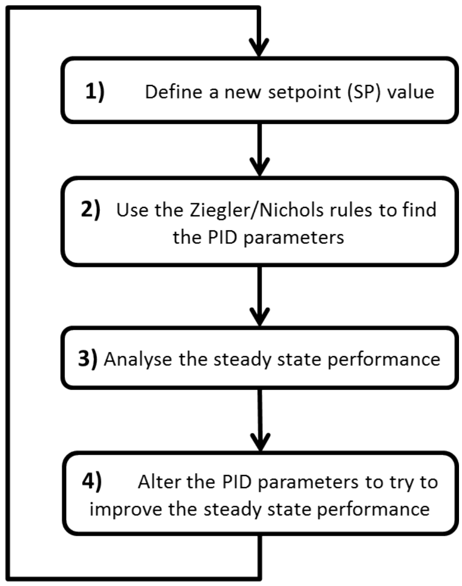


Figure 6.4: PID parameters task

## 6.2 Modelling a closed loop control

There is other option to control the automation system that can be proposed to the students. Instead of using the local *PID* and determinate the parameters without calculating the transfer function of the system, it can be created a control model in a software simulation development environment like *Matlab* or *Labview*. For this task, it would be used the remote signals to control the system.

Differently to the last task, to implement the control model of each station, it is necessary to have all the characteristics of the stations available. Not all the components of the system have their transfer function available on their manuals. Therefore, the first step to do on this task is to study all the transfer functions necessary in each station.

Having all the components transfer functions, the next step is to design the control system model of the stations. This is the main problem of this task and can be done with the help of a simulation software like *Matlab/Simulink* or *Labview*. A suggested method to design the control system model is shown in figure 6.5.

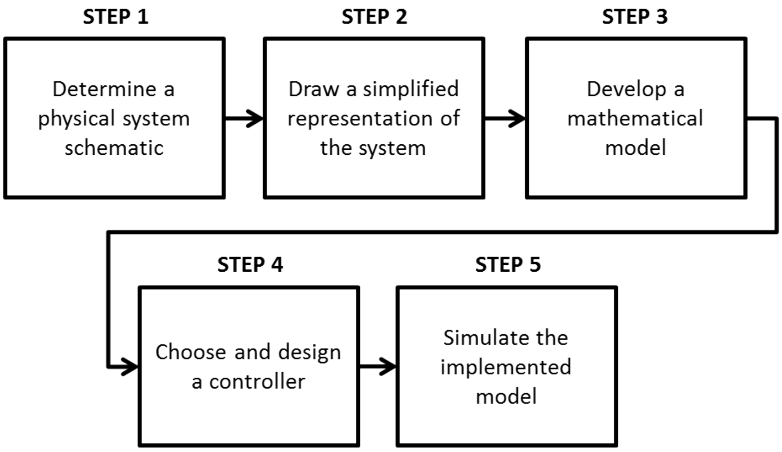


Figure 6.5: Control system design process [7]

- **STEP 1 -** Transform the system requirements to a physical model showing all the characteristics of the system;
- **STEP 2 -** After analysing the schematic, it is crucial to translate it into a simplified representation that describes the main interactions between the components;
- **STEP 3 -** Once the representation is drawn, it is used physical laws, such as *Kirchoff's* laws for electrical networks and *Newton's* laws for mechanical systems, along with simplified assumptions, to model the system mathematically. These laws lead to mathematical models that describe the relationship between the output and input (transfer function) of the dynamic system;
- **STEP 4 -** The next step is to choose a control method and design a controller. An example of a possible controller is a digital *PID*. It has basically the same functions that the one installed for the local operation mode, however it is digital and can be configured by software;
- **STEP 5 -** Finally it is possible to simulate and see if the response of the model developed is correct. If not, the model should be revised, as well as the controller.

Following in this chapter, it is described further in detail, how to create a control model for each station individually. It is important to notice that the aim of this task is to allow the students to develop the model to control the system. Therefore, the only explanation on the next sections is about the physical stations and their components. The steps two to five are left for the students to solve.

### 6.2.1 Level Station Model

The level station, as described in the first chapters, consists of two containers connected with two pipes and a pump. A representative scheme with the relevant characteristics of the station, is shown in figure 6.6. The water is conveyed from the lower container to the upper container (1) by the pump that is controlled by an analogue voltage (0 - 10 V). Then the water can flow back to the lower container by the pipe (2). The purpose of this task is to control the water level by measuring it with the ultrasonic sensor.

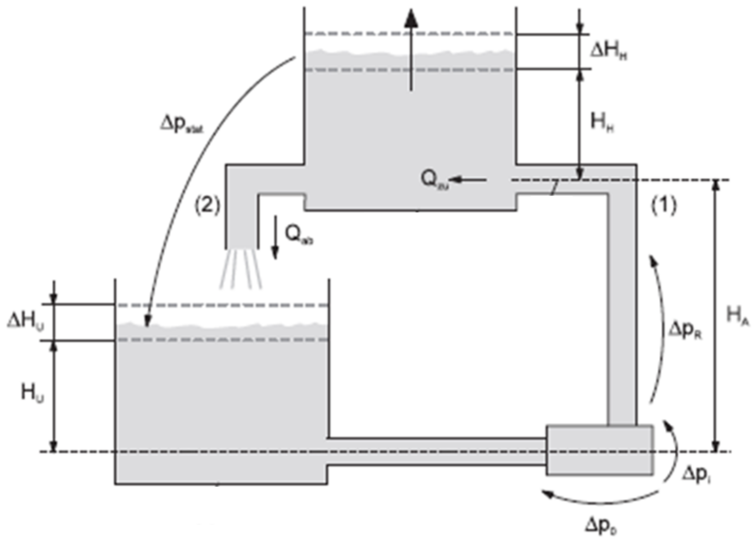


Figure 6.6: Schematic representation of level control system [8])

Abbreviations descriptions:

- P - Pressure
- Q - Volumetric flow rate
- H - Height

As it says in the beginning of this section, the first thing to do, is to study the transfer functions of the analogue components - the pump and the ultrasonic sensor.

**Ultrasonic sensor transfer function:**

The transfer function of the ultrasonic sensor is available on the *Festo* manual and can be seen on figure 6.7.

**Pump transfer function:**

It is also necessary to obtain the transfer function of the pump that is not available on the respective manual. To make this study there are two suggested ways:

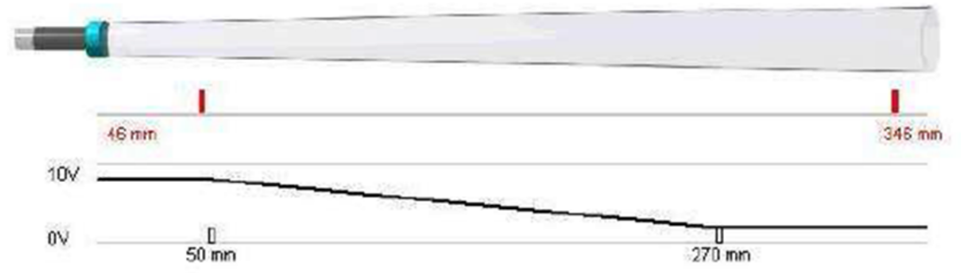


Figure 6.7: Transfer function of the ultrasonic sensor [1])

**1)** Empty the upper container and apply a voltage of 0,5 V to the pump controller. Wait one minute, measure the quantity of water inside the container and register the value. Increase 0,5 V to the previous experiment and repeat these steps. With all the values measured from 0 V to 10 V, draw the transfer function of the pump.

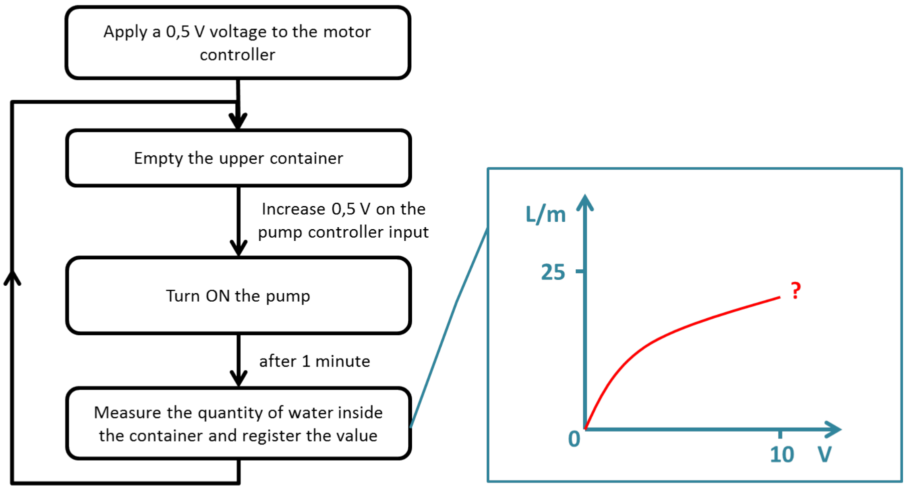


Figure 6.8: Steps to determine the pump transfer function

**2)** Instead of measuring the quantity of water inside the container after one minute, it is also possible to use the flow meter to calculate the transfer function of the pump. It is just necessary to install one flow meter after the pump, and by having the transfer function of the flow meter, it will directly give the correspondent necessary value in litres per minute.

With the transfer functions determined, the next steps can be seen in the diagram 6.5 since they are general for all the stations: draw a simplified representation of the system, develop the mathematical model, develop the control design and finally simulate the implemented model.

### 6.2.2 Temperature Station Model

The temperature station consists in one container with an electrically heatable device to increase the water temperature. To obtain a better distribution of the temperature in the container, it is available a stirrer that can agitate the water and mix it, preventing the formation of different heat levels. The purpose of this station is to control the temperature of the water inside the container by measuring the temperature with the *PT100* sensor. Since the outside temperature is lower than from the inside of the container, it has to be considered the respective losses. The scheme of the station model is represented on figure 6.9.

Abbreviations descriptions:

- T - Water temperature inside the container
- Tu - Ambient temperature (air)
- Q - Quantity of heat
- h - Filling level

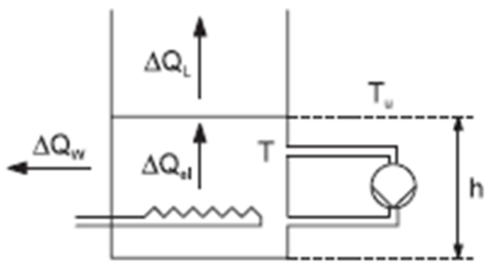


Figure 6.9: Schematic representation of temperature control system [8])

In this station it is necessary to determine the transfer functions of the heating unit, the stirrer and the temperature sensor (*PT100*). As the stirrer as the only purpose to mix the water there is no need to determine its transfer function.

**Heating unit transfer function:**

To determinate the transfer function of the heating unit, it is necessary to calculate the power of the component in relation to the input voltage. One used method is to measure the time that it takes to rise the temperature from one defined value to another (e.g. from 25°C to 27°C). To measure the temperature of the water, it can be used the temperature sensor available (*PT100*) or other. This step should be done with the input voltages from 0 to 10 V, varying 0,5 V in each experiment. The figure 6.10 represents the steps diagram of this task.

This method also can be implemented by a simulation program as *Matlab* or *Simulink* since the values of the *PT100* can be digitally used by the remote user. Using a computer would get better and faster results.

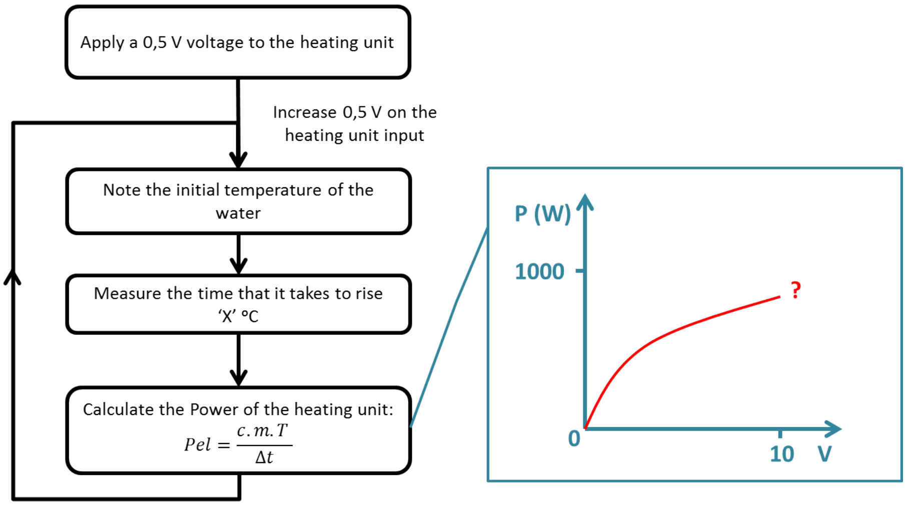


Figure 6.10: Steps to determine the heating unit transfer function

**Temperature sensor transfer function:**

To determine the transfer function of the temperature sensor (*PT100*), it would be only necessary to use other temperature sensor, with a known transfer function. Then the students would only need to note the temperature relative to each signal voltage (from 0 to 10 V).

With all the transfer functions available, as in the previous station, the next steps are shown on the diagram 6.5 (from 2 to 5).

### 6.2.3 Flow Station Model

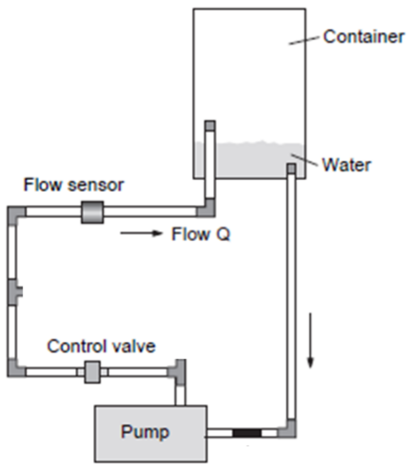


Figure 6.11: Schematic representation of flow control system [8])

The flow station consists in a container connected with one pipe from the lower part to the upper part. On this pipe it is connected a pump, a proportional valve and a flow sensor. The aim of the station is to control the water flow through this pipe (measured by the flow sensor), by partial opening or closing the proportional valve. The representative scheme is shown in figure

To determine the mathematical model of this station, it is necessary to determine the transfer functions of the proportional valve and the flow sensor.

**Proportional valve transfer function:**

The transfer function of the proportional valve is already available on the respective data sheet. It can be adjusted with two potentiometers available on the component (see [1]) as it is shown on the figure 6.12.

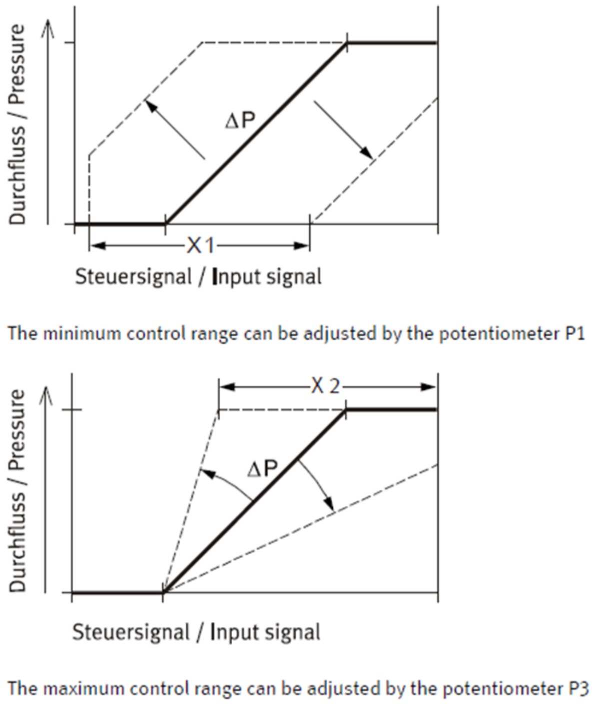


Figure 6.12: Transfer function of the proportional valve [1])

**Flow sensor transfer function:**

Since the transfer function of the flow meter is not available on the respective data sheet, it is also necessary to experimentally determine it. With the pump transfer function determined in relation to its velocity (L/min), it is just necessary to measure the flow by comparing to the pump speed. Therefore, it can be installed a flow sensor followed the pump at the level station, and gradually increase the input voltage of the pump speed controller. Then associate the respective speed value of the pump (previously acquired), to the flow sensor output (in frequency or voltage).

Having all the transfer functions determined, the students should follow the steps two to four of the diagram 6.5.

## 6.3 Conclusion

In this chapter it is described two main tasks that the students could solve with this automation system, to internalize the concepts of continuous processes control. For the correct understanding of these tasks, it is first given a brief explanation on the respective topics.

These two tasks involve two different operation modes of the system, the local and remote. If it is only desired to understand the general concept of a controller, it can be assumed that the system control model is a *black box* and determine the local *PID* parameters experimentally, by only varying the respective parameters. A method for this solution is proposed and explained on the first section of the chapter.

However, the students can also go a little further, and determine the mathematical model of the system to create a new control model for the continuous processes control. For this task it would be used the signals available for the remote user and also a simulation software to implement the control model. It is also suggested methods how to determine the transfer functions of all necessary components, since not all of them are available on the respective data sheets.

# Chapter 7

# Conclusion

The last chapter of the report starts with the review of the problem stated previously in the Chapter 3, and resumes the work done. Furthermore, the open possibilities for future work on this project. In the end, an overall conclusion contains the last words from the author, relatively to this experience.

## 7.1 Project Requisites Review and Work Done

This dissertation project aim was to develop an automation system of continuous processes for the laboratory of the Electrical Engineer Department in order to teach and train the students of Automation field on the respective topics. To make it possible, it was partially acquired a process automation system from *Festo* and the goal was to finalize and adapt it to the Department requirements. As this system is pretended to be used for didactic purposes, it was defined to create two users, local (supervision) and remote (students). The main goal of the this dissertation project was to fully develop the local mode control and also normalize all signals to be later used by a remote device. The requirements proposed and the respective work done are described below:

***Creation of a physical isolation for the sensors and actuators signals.***

During this dissertation project it was chosen and designed all the necessary hardware and respective circuits, for the full physical isolation of the required signals, remote and local. With this isolation system projected and tested, it is guaranteed the protection of the automation system for a wrong use of voltages or currents to the sensors or actuators.

***Implementation of the signals interface between the user and the Festo system.***

To allow the local user to control the automation system, it was chosen a control device that receives the inputs and generates the outputs. It was opted for a PLC that allows the user to control any operation, regarding the requires signals, and to run a program to monitor and prevent some emergency actions. It was chosen a PLC from *Beckhoff* and it supports all the requisites established. It is used to run the interlock and control the system.

***Signals adaptation.***

Each signal has its output range and they differ from each other. To use the *Beckhoff* PLC it was necessary to normalize all the signals to the required voltages. It was so, chosen, designed and tested all the components and circuits to adapt the required signals. With the implementation of the projected circuits, it is possible to connect the PLC to the final automation system and also normalize all the remote signals to facilitate the later selection of a device for the remote user control.

It was also necessary to implement a controller for the pumps in order to allow them to be controlled by the normalized analogue input signal (0 to 10 V). It was so designed and tested a control circuit that permits to control the pump with the desired signals.

***Development of a HMI to the system.***

More than being able to program the PLC to control the automation system, it is necessary to make its control easy and intuitively. It was so a requisite to implement a HMI for the proposed system.

This application was developed by *Visual Basic* and it allows the local user to monitor the analogue variables values, to control the actuators and to select which user is controlling the system. It is easy to understand as its design represents the physical structure of the automation system, making it accessible to any person to control it.

***Creation of an interlock.***

One of the main concerns when this automation system was structured, was the possibility of students make a incorrect control of the system, which could lead to situations that could damage it. To prevent these situations of happening it was programmed an application that monitors all the variables of the system and turns off some actuators it in case of a danger situation. This application is always running on the PLC. It allows the system, more than being protected to wrong physical uses, also to prevent wrong software controls.

***Suggest didactic tasks for continuous processes using the automation system developed.***

More than developing the automation system, it is important to explain how it can be used for didactic purposes on the automation field. To allow the students to understand how to use this system, it is described two main exercises that can be done in order to understand how to control a continuous process by a closed control model. By reading the chapter 6 the students have a good base to start the implementation of the control solution of the system.

## 7.2 Future Work

With the automation system developed it is indeed possible train and learn the topics related to this dissertation project. However, the practical meaning of this automation system is very little.

It only represents an interaction of the water inside the containers with different components that can change some characteristics of the system. It is not associated to a real situation.

It could be interesting to associate this automation system to a real industrial process control. There are more modules that could be added to the system, in order to represent any real control model, like a beverage industry or other industry that controls liquids.

Within the system, it could also be added some analogue channels to control all the analogue variables on the HMI. It is interesting to have all the visual access to all the characteristics of the system.

As most of the real industrial processes need to preserve all the data regarding to the processes. In order to implement it to this system, it could be also developed a database and connect it to the system. With this database, all the information desired could be saved, and the results of the experiments could be easily compared to the previous ones in order to evaluate the efficiency of the experiments.

Finally it would be interesting, after assembling the equipment, to make a full test to all the functionalities of the automation system, and observe and adjust some situations that could occur. This adjustments could be incrementing rules to the interlock of the system to alter the HMI for an easily manipulation of the system.

## 7.3 Overall Conclusion

This dissertation project was proposed to eliminate a necessity of having an automation system that could train the students on the topic of continuous processes. After an intensive study of the system, it was designed and tested all the hardware and software necessary for its operation. With the automation system working, there was also the necessity to allow two modes of operation, a local for the professors supervision, and a remote to allow the students to make their own control model.

After this dissertation project, not only the system is ready to be assembled and used by the laboratory with all the requisites proposed, but also it is available a small guide to correctly use this equipment for the training of the topics proposed. With this results, the author truly believes that, even if the automation system is still not assembled, the project and design of the automation system was successfully done and so, the goals of this dissertation project were accomplished.

# Appendix A

# HMI design

The menus of the HMI flow station and level station are represented here. The temperature menu can be seen on figure 5.8.

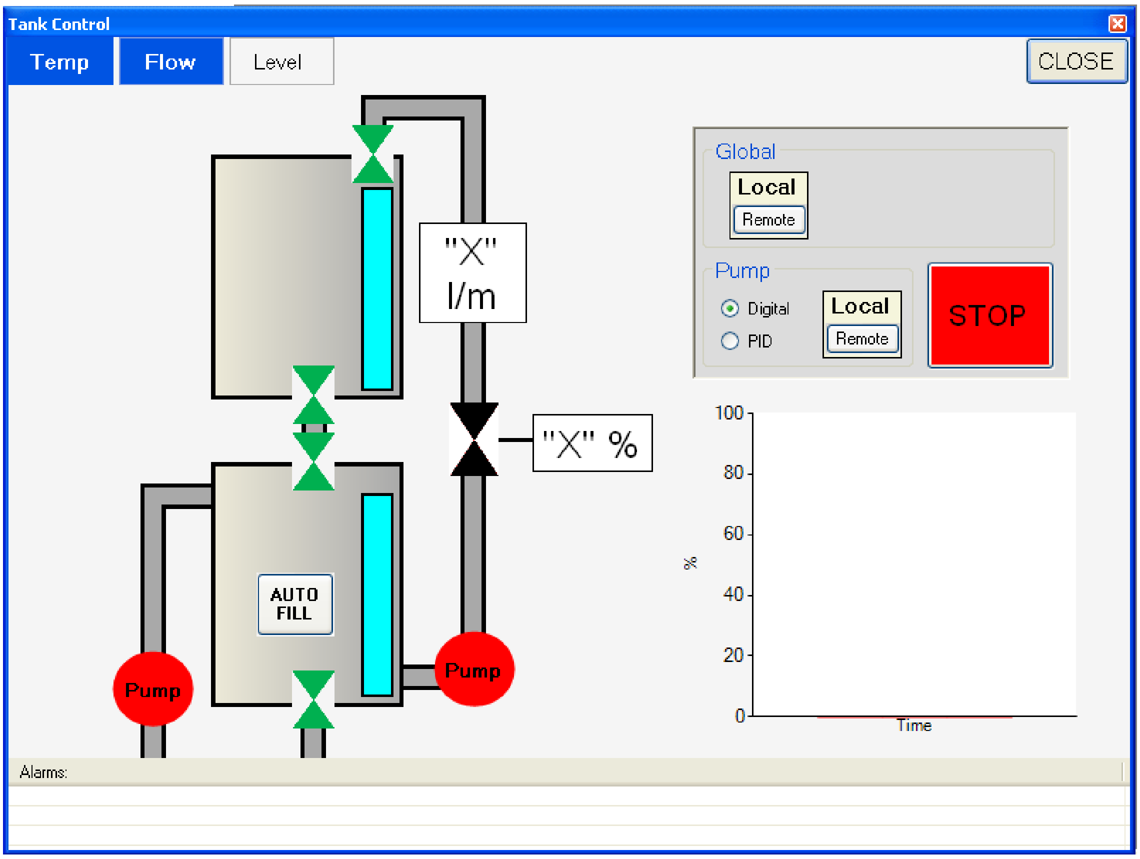


Figure A.1: HMI level menu

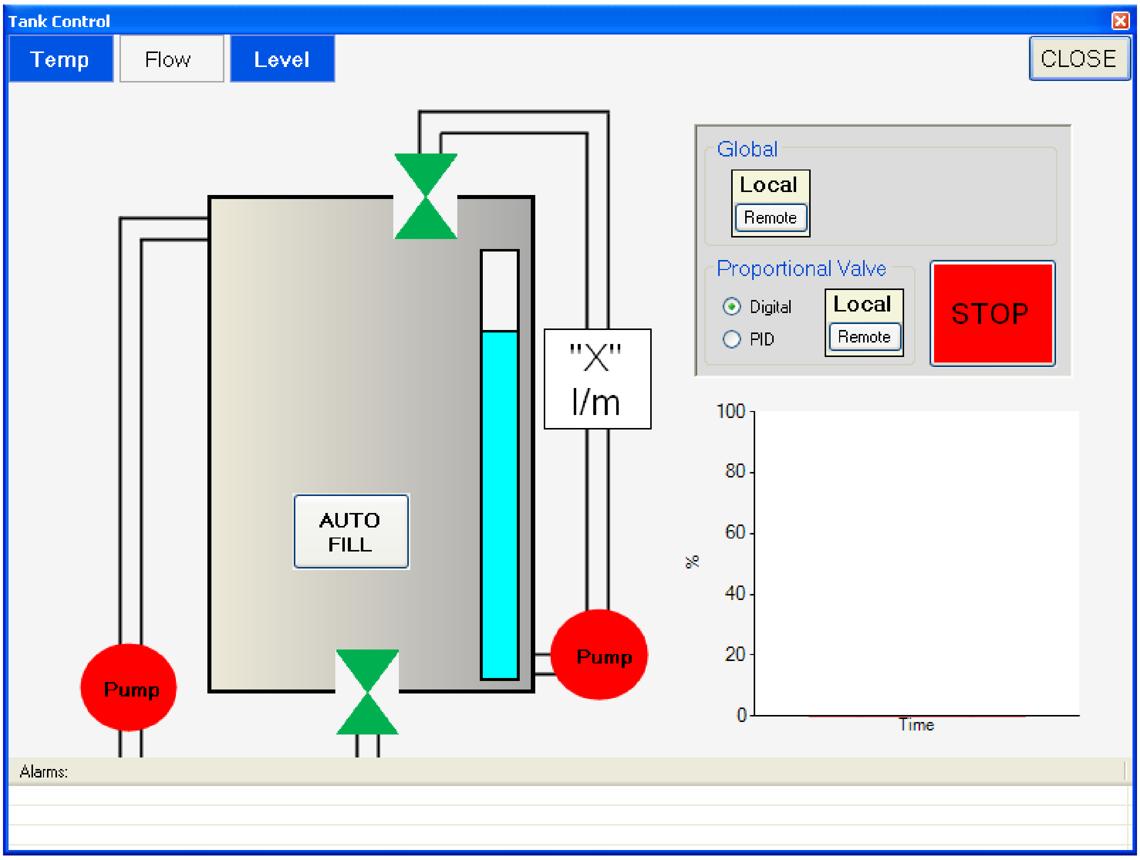


Figure A.2: HMI flow menu

# Appendix B

# Circuits

In this appendix, it is shown all the circuits implemented regarding to the digital signals of the system. Also, some figures were designed to help the comprehension of the circuits.

**Digital Actuators:**

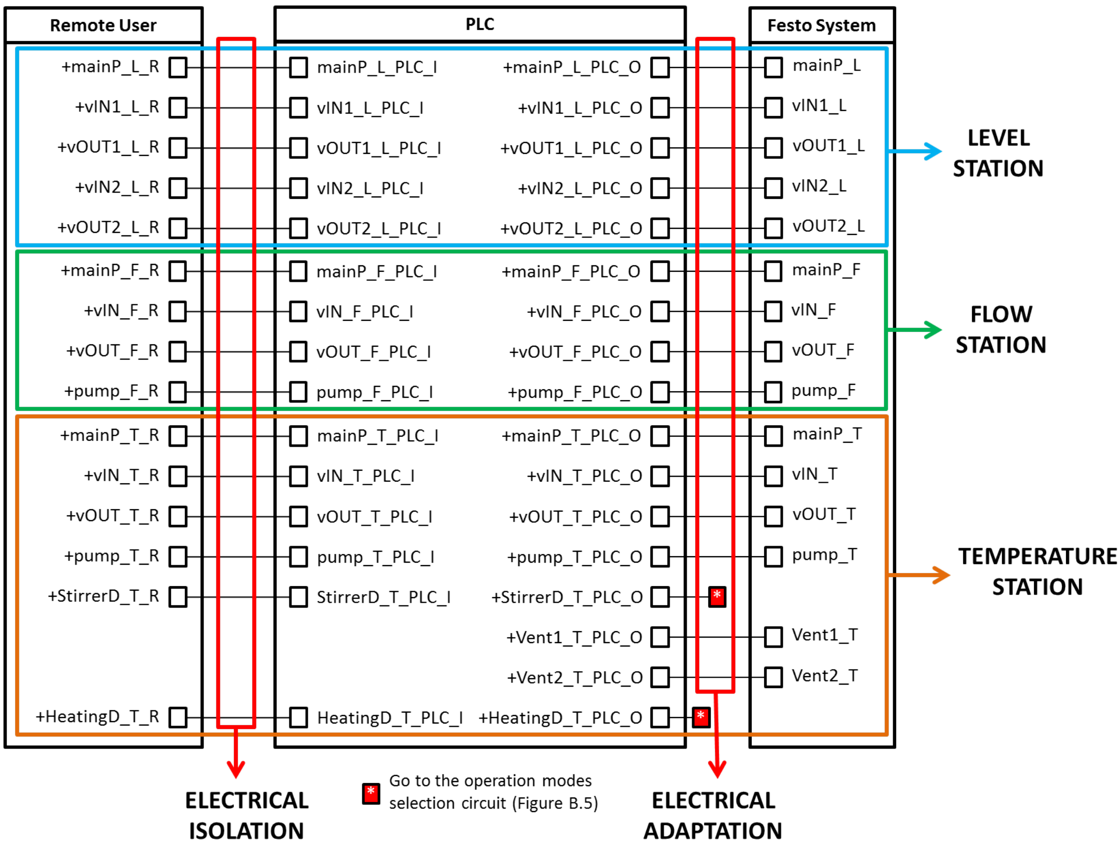


Figure B.1: Actuators digital signals scheme

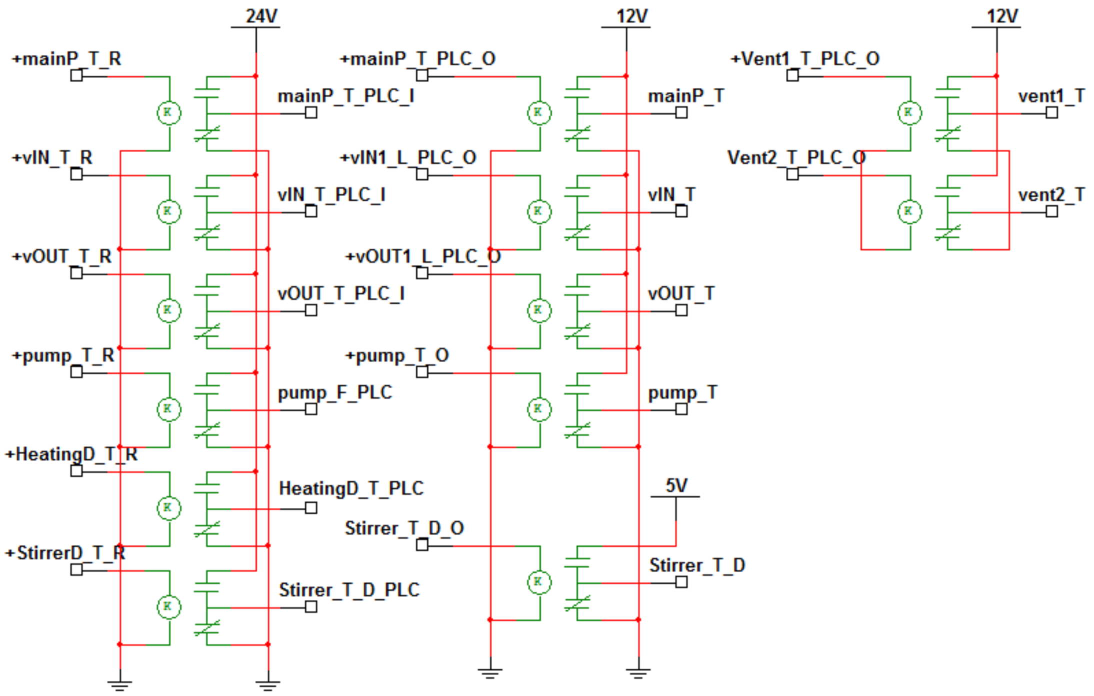


Figure B.2: Temperature digital electrical circuits - Isolation + Adaptation

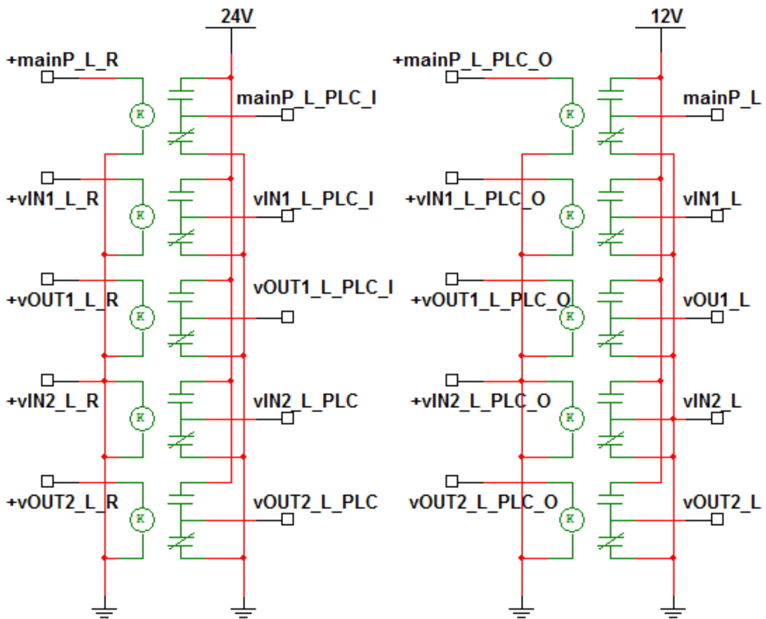


Figure B.3: Level digital electrical circuits - Isolation + Adaptation

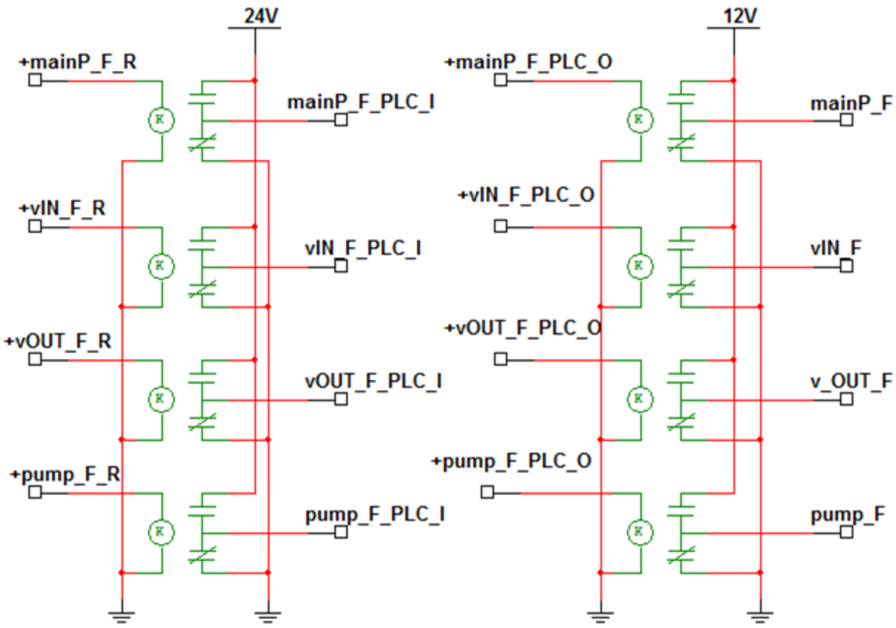


Figure B.4: Flow digital electrical circuits - Isolation + Adaptation

### Actuators analogue isolation and operation modes selection:

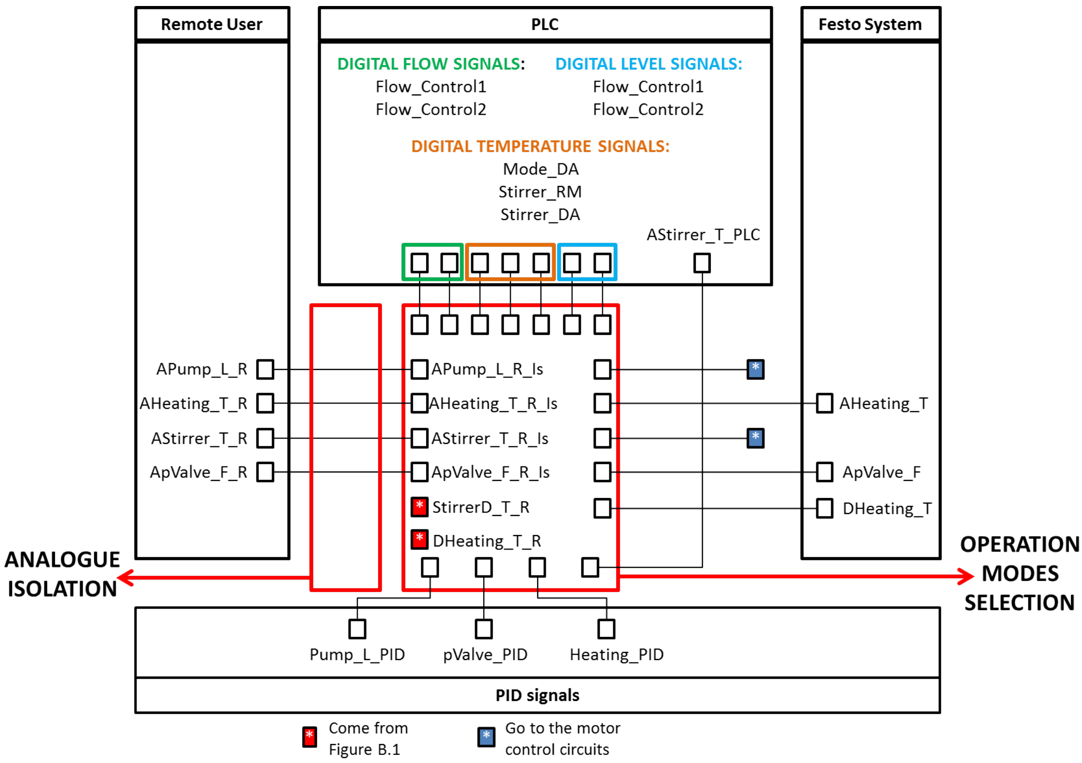


Figure B.5: Actuators analogue signals isolation and operation modes selection scheme

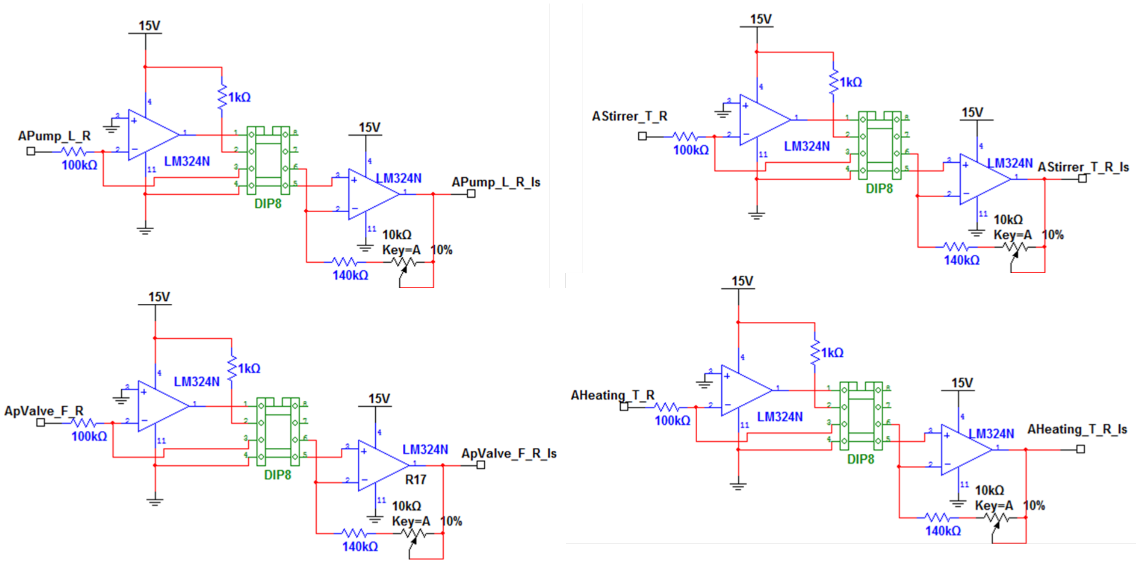


Figure B.6: Actuators analogue signals isolation circuits

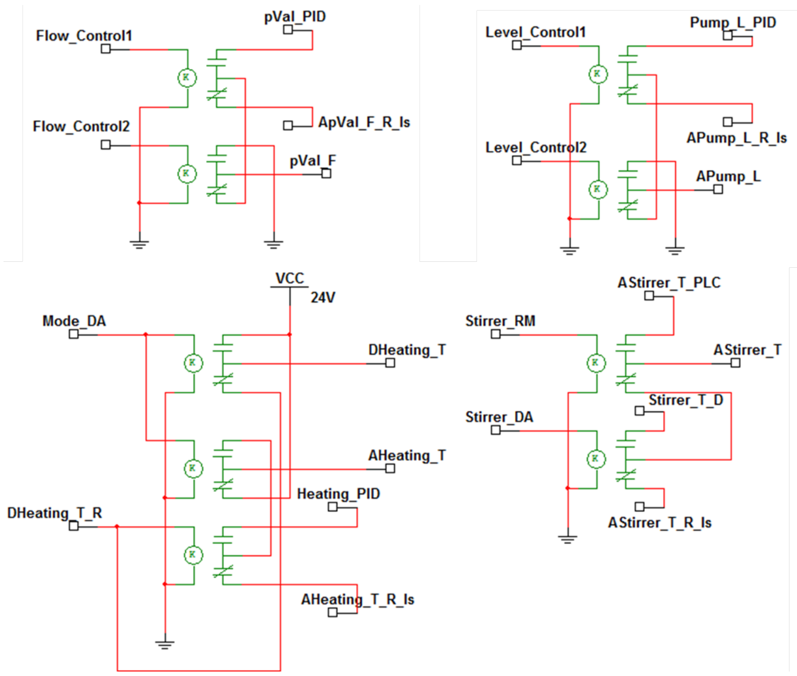


Figure B.7: Operation modes selection circuits

**Motors speed control circuits:**

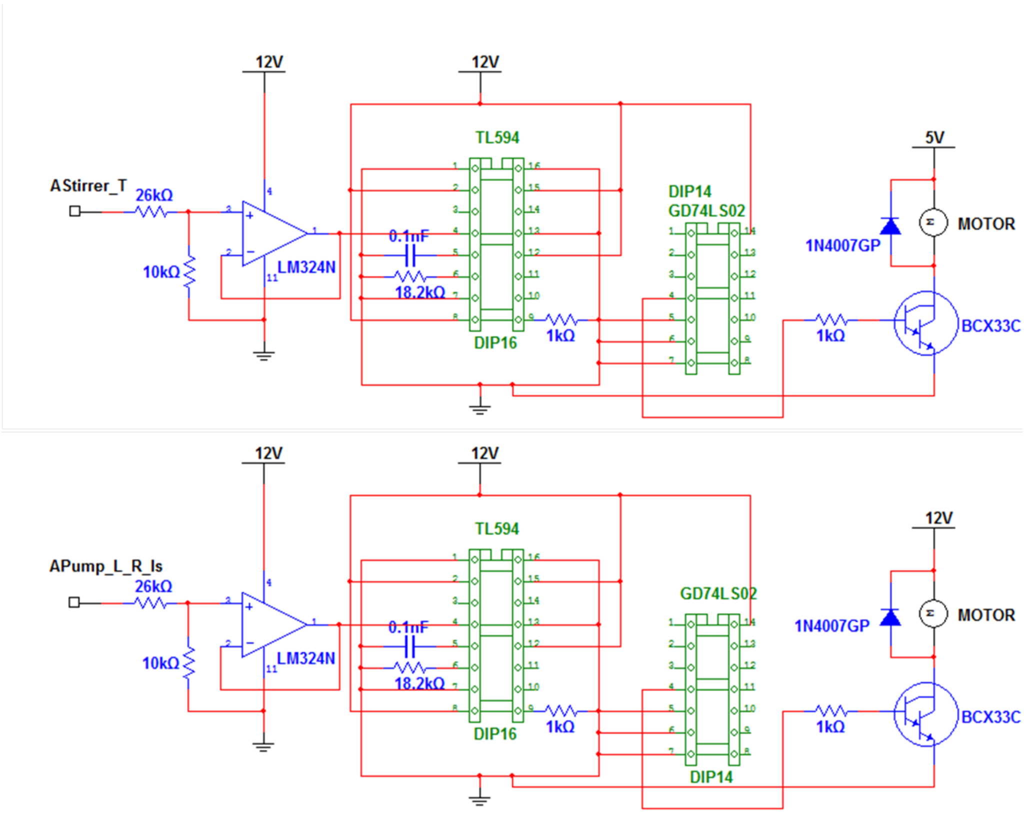


Figure B.8: Pump and stirrer speed controller circuit

**Sensors isolation and frequency to voltage converter:**

Regarding to the sensors they are all isolated for the remote user. Also there are the circuits of the flow sensor, both for frequency to voltage conversion and voltage adaptation.

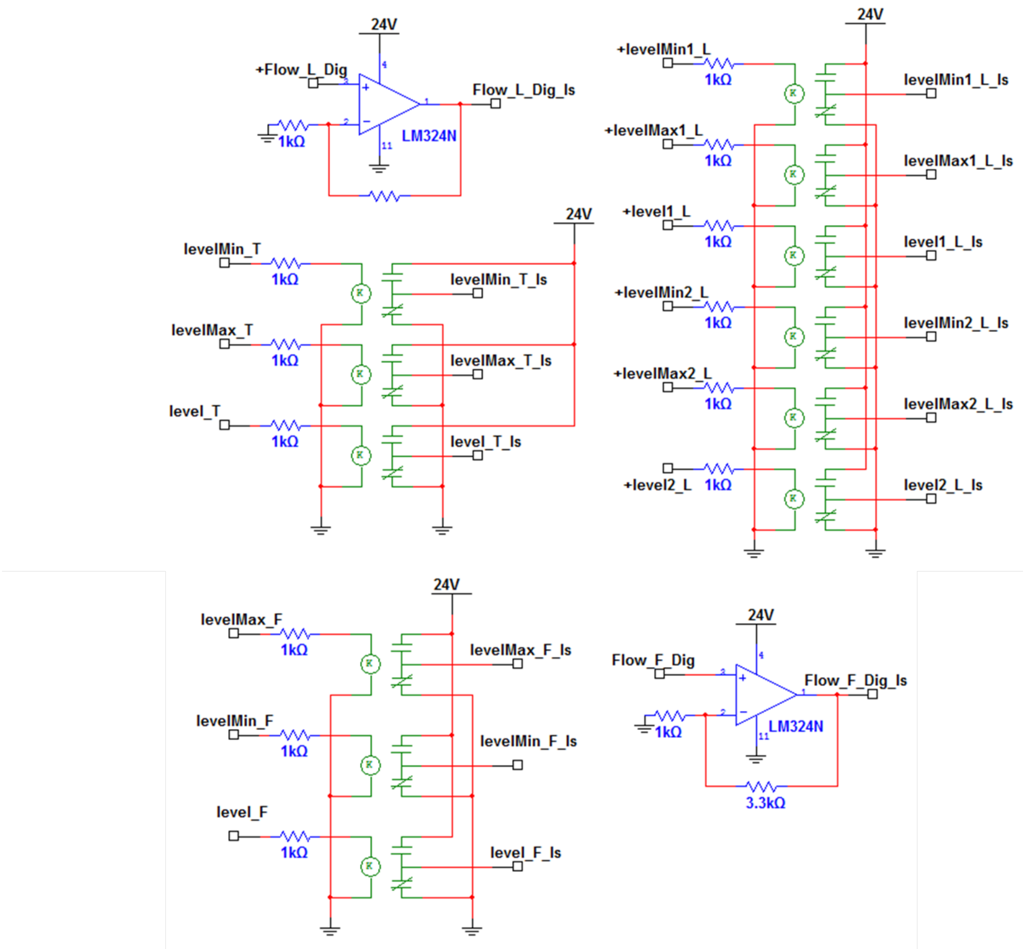


Figure B.9: Sensors isolation, adaptation and conversion circuits

# References


[1] Festo. *Festo Data Sheets*.

[2] Tim Surtell. Relays. URL: http://www.eleinmec.com/article.asp?24.

[3] Texas Instruments. *LM2907/LM2917 Frequency to Voltage Converter*, March 2013.

[4] Beckhoff. *Beckhoff TwinCAT: PC Control introduction*, 04 2001.

[5] URL: http://www.csimn.com/CSI_pages/PIDforDummies.html.

[6] Jinghua Zhong. Pid controller tuning: A short tutorial, 2006.

[7] Hussairi Bin Borhan. Design and modeling of temperature control system for house ventilation system using matlab/simulink. Master's thesis, University Teknikal Malaysia Melaka, 2007.

[8] E. v. Terzi H. Bischoff*, D. Hofmann*. *Process Control System Control of temperature, flow and filling level*. Festo Didactic GmbH & Co., 1997.

[9] Festo Didatic. *Process Control System, Control of Temperature, flow and filling level*, 1997.

[10] Mr. Naqui Anwer. Programmable logic controller (plc) : Definition. URL: http://makox.com/plc-scada/1-introduction-of-plc-scada/programmable-logic-controller-plc-definition/.

[11] PC Mag. URL: http://www.pcmag.com/encyclopedia_term/0,,t=&i=49992,00.asp.

[12] Beckhoff. *Beckhoff Trainig Series Module 1: Introduction to TwinCAT*, 2007. URL: http://pt.scribd.com/doc/111907282/BECKHOFF-TwinCAT-Training-en.

[13] URL: http://infosys.beckhoff.com/english.php?content=../content/1033/tcsample_vb/html/tcadsocx_vb_setup.htm&id=.

[14] Microsoft .net framework 4, 2 2011. URL: http://www.microsoft.com/en-us/download/details.aspx?id=17851.

[15] Sigurd Skogestad. Probably the best simple pid tuning rules in the world. *J. Process Control*, page 1, 2001.

[16] Tomas B. Co. Ziegler-nichols method, 2 2004. URL: http://www.chem.mtu.edu/~tbco/cm416/zn.html.

[17] Vishay. *Linear Optocoupler, High Gain Stability, Wide Bandwidth*, October 2012.

[18] VISHAY SEMICONDUCTORS. *Designing Linear Amplifiers Using the IL300 Optocoupler*, March 2012.

[19] Beckhoff. *PC Control introduction*, April 2001.